

Maximal assortative matching for real-world network graphs, random network graphs and scale-free network graphs

Natarajan Meghanathan¹

Received: 24 October 2015 / Accepted: 14 March 2016 / Published online: 30 March 2016
© The Author(s) 2016. This article is published with open access at Springerlink.com

Abstract We define the problem of maximal assortativity matching (MAM) as a variant of the maximal matching problem wherein we want to maximize the similarity between the end vertices (with respect to any particular measure for node weight) constituting the matching. The MAM algorithm (with a targeted assortative index value of 1) works on the basis of the assortative weight of an edge, defined as the product of the number of uncovered adjacent edges and the absolute difference of the weights of the end vertices of the edge. The MAM algorithm prefers to include the edge with the smallest assortativity weight (the assortative weight of the edges is updated for each iteration) until all the edges in the graph are covered. We show that the MAM algorithm can be easily adapted to be used for maximal dissortative matching (MDM) with a targeted assortative index of -1 for the matching as well as for maximal node matching (MNM) algorithm to maximize the percentage of nodes matched. We illustrate the execution of the MAM, MDM and MNM algorithms on complex network graphs such as the random network graphs and scale-free network graphs as well as on real-world network graphs and analyze the tradeoffs.

Keywords Maximal matching · Assortative matching · Dissortative matching · Assortativity index · Complex networks · Independent edge set · Node similarity · Random networks · Scale-free networks

1 Introduction

A matching M for a graph $G = (V, E)$ is a subset of the edges E such that no two edges in M have a common vertex. A maximal matching is a set of independent edges such that the inclusion of any additional edge to the set violates the property of matching (no common vertex between any two edges of the set). A matching for a graph is said to be maximum if every vertex in the graph could be matched with another vertex of the graph through a set of edges such that no two edges in the set have a common vertex. There may exist maximal matching of various sizes for the vertices of a graph; but, every maximal matching need not be a maximum matching; on the other hand, a maximum matching of the vertices in a graph is the largest possible maximal matching for the vertices of the graph. Accordingly, we refer to the maximum matching problem as a problem of finding the largest set of independent edges whose end vertices form the non-overlapping node pairs such that the maximum number of node pairs is $\frac{V}{2}$ if the number of vertices V is even and is $\frac{V}{2} - 1$ if the number of vertices V is odd.

A well-known algorithm for finding the maximum set of independent edges for maximum node matching in arbitrary network graphs is the Blossom algorithm [1] of time-complexity $O(V^4)$ on a graph of V vertices. Several improvements (e.g., [2]) to the Blossom algorithm have been proposed in the literature. A weakness of all these algorithms is that in pursuit of maximum node matching, little consideration is given to the similarity between the vertices that are matched. As observed in the simulations of this paper, a maximum or maximal node matching of the vertices in a complex network graph need not match vertices of comparable node weight (e.g., node degree). The motivation for the research presented in the paper stems from this observation. We want to determine a maximal matching (need not

✉ Natarajan Meghanathan
natarajan.meghanathan@jsums.edu

¹ Jackson State University, Jackson, MS 39217, USA

be the maximum matching, but close enough to the maximum matching) of the vertices that are very similar to each other (or very dissimilar from each other). This amounts to maximizing (or minimizing) a metric called the assortativity index of the edges that constitute the matching. Until now in the literature for complex network graphs, assortativity has been considered only at the network level [3] and node level [4,5], but not with respect to the matching of the vertices. Ours is the first paper in this direction.

The assortativity index of a set of edges (with respect to any particular measure of node weight-like the node degree) is a quantitative measure of the similarity between the end vertices of the edges that are part of the set [6]. The assortativity index values can range from -1 to 1 . If the assortativity index of a set of edges calculated with respect to a particular measure of node weight is close to 1 , then it implies the end vertices of the edges that form the set are very similar to each other with respect to the particular measure of node weight (for example, a high-degree vertex matched to another high-degree vertex, a low-degree vertex matched to another low-degree vertex, etc). If the assortativity index is close to 0 , then the pairing of the vertices in the edge set is arbitrary with respect to the node weight. On the other hand, if the assortativity index of the set of the edges with respect to a measure of node weight is close to -1 , then it implies that most of the node pairs constituting the edge set are not similar to each other with respect to the node weight (for example, if node degree is used as the node weight, then an assortativity index of -1 of a set of edges implies that most of the node pairings in this set involve a high degree vertex matched to a low degree vertex and vice-versa).

For social networks and other complex real-world networks where peer-to-peer interaction and collaboration are preferred, it might be useful to pair vertices that are very similar (or very dissimilar) to each other as part of a maximal matching of the vertices in the network. A maximal matching that is arbitrary with respect to the weight of the vertices being matched need not be preferred in social networks. For example, a researcher who already has some accomplishments to his/her credit may want to pair with another researcher who also has a similar research profile (say quantified in terms of the number of peer-reviewed publications in a research area) so that they can mutually collaborate and benefit from each other. On the other hand, a newly joining researcher to a social forum (like researchgate.net or linkedin.com) may want to pair with an accomplished researcher. If each node in a social network can be matched with only one another node at a time, then it is imperative to match the nodes that are either dissimilar to each other or similar to each other (depending on the application of interest); an arbitrary matching of the vertices in a social network may not be of any practical benefit. To the best of our knowledge, we have not come across a maximal matching algorithm that maximizes

the assortativity index (for matching nodes that are similar to each other) or minimizes the assortativity index (for matching nodes that are very different from each other) in complex network graphs.

In this paper, we propose a maximal matching algorithm that can be used to maximize or minimize the assortativity index of the edges constituting the matching determined in complex network graphs where the nodes have weights (the smaller the difference in the node weights, the more similar are the nodes and vice-versa). An edge that is part of a matching is said to cover itself as well as cover the edges adjacent to it in the original graph and these edges cannot be part of the matching. We define a metric called the assortativity weight of an edge as the product of the number of uncovered edges adjacent to the edge in the graph and the absolute value of the difference in the weights of the end vertices constituting the edge. The maximal matching algorithm for maximizing the assortativity index (hereafter, referred to as the maximal assortative matching algorithm, MAM) prefers to include edges that have lower assortativity weight as part of the matching. The algorithm runs in iterations. In each iteration, we determine a ranking of the uncovered edges in the graph based on the assortativity weight metric defined above and choose the edge with the smallest value for the assortativity weight metric and include it among the edges constituting the matching. We continue the iterations until all edges in the graph are covered. An edge with the smallest value for the assortativity weight is likely to have fewer adjacent edges as well as comprise of end vertices with close-enough node weights. Our hypothesis is that by choosing such edges with smaller values for the assortativity weight, for graphs that are sufficiently dense, we can simultaneously maximize the assortativity index of the matching as well as maximize the number of edges chosen as part of the matching. The proposed algorithm would be very useful for matching vertices in social networks and other real-world networks for peer-to-peer interaction and collaboration.

Ours will be the first such algorithm to determine a maximal matching of the vertices based on the notion of assortative weight of the edges and does not use the notion of augmenting paths [7], as used by most of the existing matching algorithms. We evaluate the performance of the proposed maximal assortative matching (MAM) algorithm on six real-world network graphs whose degree distribution range from Poisson (random networks) [8] to Power-law (scale-free networks) [9] as well as run the algorithm on complex networks simulated from theoretical models such as the Erdos–Renyi model (for random networks) [10] and Barabasi–Albert model (for scale-free networks) [11]. We observe the MAM algorithm to determine a maximal matching of the nodes (the end vertices of each node pair are similar to each other) and the overall assortativity index of the matching is significantly larger than a matching of the

nodes determined with the objective of just maximizing the number of nodes matched.

The focus of the paper is on presenting the proposed maximal assortative matching algorithm for maximizing the assortativity index of the matching. Towards the end of the paper, we also show that the algorithm can be used to minimize the assortativity index of the matching (maximum dissortative matching) by simply including the edges with the largest assortativity weight as part of the matching in each iteration (no other modifications are required). The rest of the paper is organized as follows: Sect. 2 presents the maximal assortative matching (MAM) algorithm for an arbitrary graph and discusses its flexibility to be used as a maximal matching algorithm for maximizing the number of nodes matched (hereafter referred to as the maximal node matching algorithm, MNM). Section 3 presents the results of the execution of the MAM and MNM algorithms on real-world network graphs with degree distribution ranging from Poisson to Power-law. Sections 4 and 5 present the results of the execution of the MAM and MNM algorithms on random networks generated according to the Erdos–Renyi model with the node degree as node weights and random node weights, respectively. Section 6 presents the results of the execution of the MAM and MNM algorithms on scale-free networks generated according to the Barabasi–Albert model. Section 7 presents a modification of the maximal assortativity matching algorithm (referred to as the maximal dissortative matching algorithm, MDM) to determine a matching with an objective of minimizing the assortativity index and presents results of execution of the MDM algorithm on network graphs considered in Sects. 3–6. Section 8 discusses related work and highlights the contribution of the research presented in this paper. Section 9 concludes the paper. Throughout the paper, the terms ‘node’ and ‘vertex’, ‘link’ and ‘edge’ as well as ‘pair’ and ‘match’ are used interchangeably. They mean the same.

2 Maximal assortative matching (MAM) Algorithm

2.1 Network model and definitions

We model the input network graph $G = (V, E)$ as a set of vertices V and undirected edges E wherein each vertex $v \in V$ has a weight $w(v) \in \mathbb{R}$. We say an edge (p, q) is adjacent to an edge (r, s) if $p, q, r, s \in V$ and either $p = r$ or $p = s$ or $q = r$ or $q = s$. That is, two edges (p, q) and (r, s) are said to be adjacent to each other if they have one common end vertex. The degree of a vertex $u \in V$ is the number of edges incident on u (i.e., the number of edges that have vertex u as one of the two end vertices). Though the edges are undirected, for the sake of discussion, we refer to the first

vertex (vertex u) indicated in an edge (u, v) as the upstream vertex and the second vertex (vertex v) indicated in an edge (u, v) as the downstream vertex. Also, since the edges are undirected, we conveniently adopt a convention to represent the edges: the ID of the upstream vertex of an edge (u, v) is always less than the ID of the downstream vertex of the edge (i.e., $u < v$).

A matching M of the vertices in a graph $G = (V, E)$ is a subset of the set of edges E such that no two edges in the set M have a common end vertex [7]. We refer to the edges that are part of a matching as a set of independent edges. A maximal matching is a set of independent edges of the graph such that the inclusion of an additional edge to the set violates the property of matching (i.e., no two edges of a matching have a common end vertex) [12]. A maximum node matching is the largest set of independent edges such that the number of vertices that could be paired is the maximum. The maximum node matching for a graph is one of the maximal matchings of the graph, but not vice-versa. We refer to maximal node matching as a matching determined with the objective of maximizing the number of nodes matched, but the size of the matching is not guaranteed to be that of the maximum node matching.

For a set of edges M constituting a matching of the vertices V in the graph G , the assortativity index of M is a quantitative measure of the similarity (or equivalently the dissimilarity) of the end vertices of the edges in M [6]. The assortativity index for a set M of edges (AI_M) with respect to the node weights $w(v)$ for every vertex $v \in V$ is calculated using the formula (1) given below, where \bar{U} and \bar{D} are, respectively, the average weight of the upstream and downstream vertices of the edges constituting the set M .

$$AI_M = \frac{\sum_{(p,q) \in M} [w(p) - \bar{U}] [w(q) - \bar{D}]}{\sqrt{\sum_{(p,q) \in M} [w(p) - \bar{U}]^2} \sqrt{\sum_{(p,q) \in M} [w(q) - \bar{D}]^2}}; \quad \bar{U} = \frac{1}{|M|} \sum_{(p,q) \in M} w(p); \quad \bar{D} = \frac{1}{|M|} \sum_{(p,q) \in M} w(q) \quad (1)$$

The assortativity index for a set of edges M (AI_M) could range from -1 to 1 . If AI_M is close to 1 , it indicates that the end vertices of the edges in M are similar to each other with respect to the node weight used in calculating the assortativity index. If AI_M is close to -1 , it indicates that the end vertices of the edges in M are very much different from each other with respect to the node weight used in calculating the assortativity index. If AI_M is close to 0 , it indicates that the matching of the vertices is quite arbitrary with respect to the node weight used in calculating the assortativity index.

Fig. 1 Pseudo-code for the maximal assortativity matching (MAM) algorithm

Input: Graph $G = (V, E)$, where weight $w(v) \in \mathbb{R}$ is the weight of node v for every $v \in V$
Output: Maximal Assortativity Matching, *MAM*
Auxiliary Variables: *UncoveredEdges*
Initialization: *UncoveredEdges* = E ; *MAM* = ϕ
Begin MAM Algorithm
1 **FindAssortativityWeights**(*UncoveredEdges*)
2 **while** (*UncoveredEdges* $\neq \phi$) **do**
3 Edge $(u, v) = \left\{ (p, q) \mid \text{Min}[\text{AssortativityWeight}(p, q)] \right\}_{(p, q) \in \text{UncoveredEdges}}$
4 *MAM* = *MAM* $\cup \{(u, v)\}$
5 *UncoveredEdges* = *UncoveredEdges* - $\{(u, v)\}$
6 **RemoveEdges**((u, v) , *UncoveredEdges*)
7 **FindAssortativityWeights**(*UncoveredEdges*)
8 **end while**
9 **return** *MAM*
End MAM Algorithm

2.2 Assortativity weight of an edge and hypothesis

We say an edge (u, v) included in a matching covers itself as well as covers the edges that are adjacent to it. An uncovered edge is an edge in the graph that is not yet covered by an edge in the matching. We define the assortativity weight of an edge (u, v) to be the product of the number of uncovered edges that are adjacent to it and the absolute value of the difference in the weights of the end vertices u and v . The number of uncovered edges adjacent to an edge (u, v) is the number of uncovered edges incident on each of the end vertices u and v . The proposed maximal assortative matching (MAM) algorithm for maximizing the assortativity index of the matching (to be as close to 1 as possible) proceeds in iterations. In each iteration, we include the uncovered edge with the smallest assortativity weight as one of the edges constituting the matching and consider it to have covered itself as well as its adjacent edges. Our hypothesis is that by giving preference to edges with lower assortativity weight (as defined above), we choose edges whose end vertices have weights that are as close as possible (primary objective) as well as be able to maximize the number of independent edges that are chosen to be part of the matching (secondary objective). As observed in Sects. 3–7, it may not be possible to simultaneously accomplish both the above objectives (especially for sparse graphs); there could exist a tradeoff—as the primary objective of the MAM algorithm is to give preference to edges whose end vertices have close-enough weights, the size of the maximal assortative matching could be less than the size of a maximal node matching (MNM). On the other hand (as observed in the results of Sects. 3–6), the assortativity index of an MAM for a graph could be significantly larger than the assortativity index of an MNM for the same graph. Note that the MAM algorithm could be easily transformed to an MNM algorithm by setting the assortative weight of an edge (u, v) to be simply the number of uncovered edges

adjacent to it and giving preference to edges that have lower assortativity weight for inclusion to the MNM.

2.3 Description of the algorithm for maximal assortative matching

The MAM algorithm employs a greedy strategy and at the beginning of each iteration, the algorithm chooses the uncovered edge with the smallest assortativity weight. The pseudo-code for the algorithm to determine maximal assortative matching (MAM) is outlined in Fig. 1; the pseudo-code for the two sub routines used in the algorithm is given in Fig. 2. The algorithm maintains the set of uncovered edges (*UncoveredEdges*) that are yet to be covered by an edge in the MAM. The set *UncoveredEdges* is initialized to the set of all edges E for the input graph G .

To start with, the assortativity weight of the edges in the set *UncoveredEdges* is determined and the edge (u, v) that has the smallest assortativity weight among the edges in *UncoveredEdges* is selected for inclusion in the MAM. An edge (u, v) selected for inclusion to the MAM is said to cover itself as well as cover its adjacent edges; accordingly, all these newly covered edges are removed from the set *UncoveredEdges*. The assortativity weight of the edges in the updated set of *UncoveredEdges* is recalculated and the edge with the smallest assortativity weight is selected for inclusion in the MAM. The above procedure is repeated as a sequence of iterations until the set of *UncoveredEdges* is empty. At this stage, we have found a maximal matching of the vertices in the graph.

The run-time complexity of the MAM algorithm depends on the time complexity to update the set of *UncoveredEdges* in each iteration. As the algorithm proceeds, with each edge added to the MAM, we expect the size of the set of *UncoveredEdges* to reduce significantly. For optimal run-time, we suggest maintaining the set of *UncoveredEdges* as a minimum heap [7] that can be constructed in $O(E)$ time for the

Fig. 2 Pseudo-code for the subroutines used by the maximal assortativity matching algorithm

```

Subroutine FindAssortativityWeights(UncoveredEdges)
  for every edge  $(u, v) \in \text{UncoveredEdges}$  do
     $\text{numUncoveredAdjacentEdges}(u, v) = 0$ 
    for every edge  $(u, p) \in \text{UncoveredEdges}$  AND  $p \neq v$  do
       $\text{numUncoveredAdjacentEdges}(u, v) = \text{numUncoveredAdjacentEdges}(u, v) + 1$ 
    end for
    for every edge  $(v, q) \in \text{UncoveredEdges}$  AND  $q \neq u$  do
       $\text{numUncoveredAdjacentEdges}(u, v) = \text{numUncoveredAdjacentEdges}(u, v) + 1$ 
    end for
     $\text{AssortativityWeight}(u, v) = \text{numUncoveredAdjacentEdges}(u, v) * |w(u) - w(v)|$ ,
    where  $w(u)$  and  $w(v)$  are the weights of nodes  $u$  and  $v$  respectively
  end for
End Subroutine

Subroutine RemoveEdges(Edge  $(u, v)$ , UncoveredEdges)
  for every edge  $(u, v) \in \text{UncoveredEdges}$  do
    for every edge  $(u, p) \in \text{UncoveredEdges}$  AND  $p \neq v$  do
       $\text{UncoveredEdges} = \text{UncoveredEdges} - \{(u, p)\}$ 
    end for
    for every edge  $(v, q) \in \text{UncoveredEdges}$  AND  $q \neq u$  do
       $\text{UncoveredEdges} = \text{UncoveredEdges} - \{(v, q)\}$ 
    end for
  end for
End Subroutine

```

E edges of the graph. Each update to the minimum heap (like removing an edge or updating the assortativity weight of an edge) takes $O(\log E)$ time. The MAM algorithm can run at most for $V/2$ iterations for a graph of V vertices. During each such iteration, there would have to be at most E updates to the heap (one update or removal for each edge, depending on the case), incurring a worst-case time complexity of $O(E \log E)$ per iteration. Considering that there could be at most $V/2$ iterations, the overall run-time complexity of the MAM algorithm is $O(EV \log E)$. For sparse graphs ($E = O(V)$), the run-time complexity of the MAM algorithm would be $O(V^2 \log V)$; for dense graphs ($E = O(V^2)$), the run-time complexity of the MAM algorithm would be $O(V^3 \log V)$.

2.4 Algorithm for maximal node matching (MNM)

The MAM algorithm can be easily adapted to be used as an algorithm for maximal node matching (MNM). In this pursuit, the assortativity weight of an uncovered edge (u, v) in Subroutine *FindAssortativityWeights* could be simply set as the number of uncovered adjacent edges of the edge (u, v) . There would be no other change required in the pseudo-code of the MAM algorithm, as illustrated in Figs. 1 and 2. This modification would be sufficient to maximize the number of independent edges that can get selected as part of the matching. Note that the MNM algorithm is independent of the node weights as the assortativity weight for an edge is measured simply to be the number of uncovered adjacent edges; thus, the maximal node matching obtained using the MNM algorithm for a given graph would be the same irrespective of

the criterion used for node weights. By iteratively giving preference to including edges that have the lowest number of uncovered edges into the MNM, we are maximizing the chances of accommodating as many independent edges as possible into the MNM and it would be apt to call such a matching as maximal node matching. Our hypothesis is further vindicated by the results observed in Sects. 3–6.

2.5 Example for maximal assortative matching and maximal node matching

Figure 3 presents an example to illustrate the execution of the maximal assortative matching algorithm on a graph wherein the node weights are random numbers generated in the range 0–1. All the edges in the input graph and the initialization graph are uncovered edges. The initialization graph displays the assortative weight of the edges as a tuple. For an edge (u, v) , we indicate a tuple representing (number of uncovered adjacent edges and the absolute value of the difference in the node weights of the end vertices u and v) as well as the assortativity weight of the edge, which is the product of the two entries in the tuple. In the first iteration, the algorithm encounters a tie between edges (3, 6) and (4, 7)—both of which have the lowest assortative weight of 0.6; the algorithm breaks the tie arbitrarily by including edge (3, 6) to the maximal assortative matching (MAM). As part of the inclusion of the edge (3, 6) into the MAM, all its adjacent edges are considered to be covered and are removed from the graph. We reevaluate the assortativity weight of the uncovered edges in the graph; edge (4, 7) with the currently lowest assortative weight of 0.3 is the second edge to be picked for inclusion to the MAM

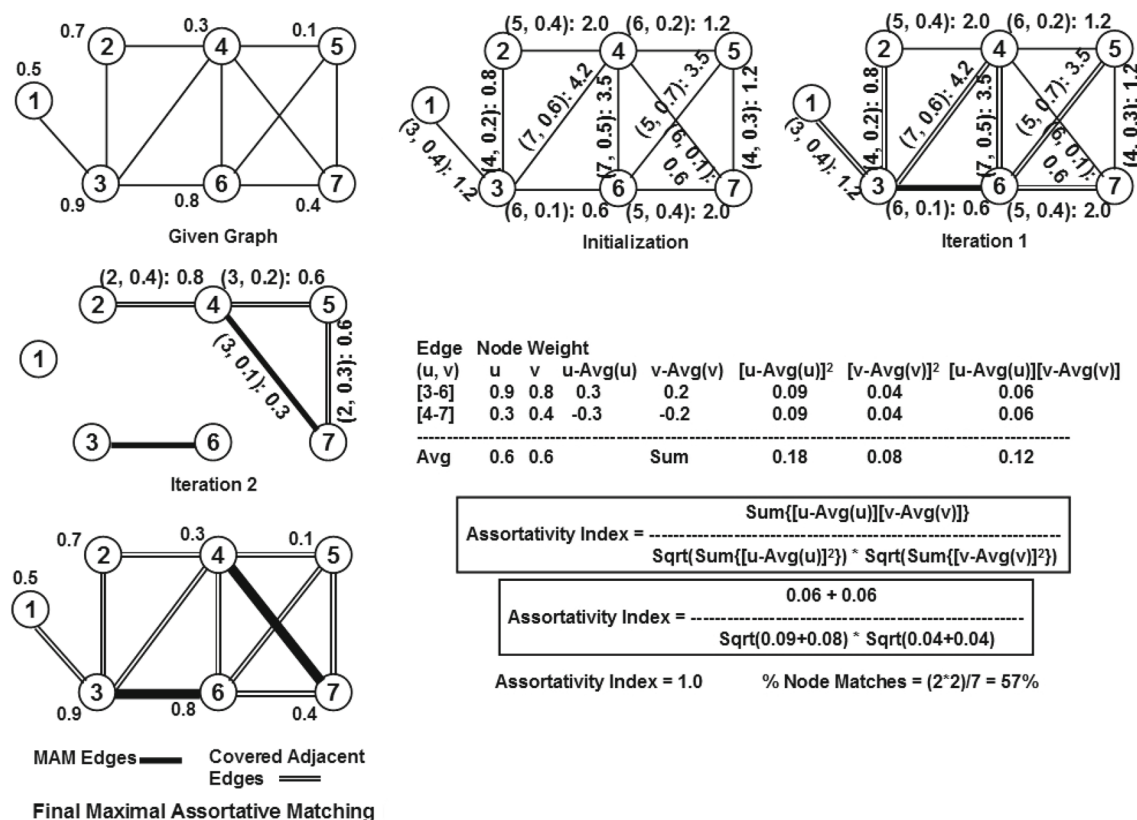


Fig. 3 Example to illustrate the execution of the algorithm for maximal assortative matching

and all its adjacent edges are removed from the graph. At the end of the second iteration, all edges in the graph are either in the MAM or covered by an edge in the MAM. The node weights of the end vertices that are included into the MAM are (0.9, 0.8) and (0.3, 0.4) for the edges (3, 6) and (4, 7), respectively. The difference in the node weights of the end vertices for both the edges in the MAM is the bare minimum that we could get for the input graph considered (as one can notice, all the nodes in the input graph have unique weights). The % of nodes matched in the MAM is $4/7 = 57\%$ and the assortative index of the matching (based on node weights) is 1.0; the calculations are illustrated as part of Fig. 3.

Figure 4 presents an example to illustrate the execution of the maximal node matching (MNM) algorithm on the same graph used in Fig. 3. The initialization graph displays the assortativity weight of the edges and in the case of MNM, it is simply the number of uncovered adjacent edges. The first edge to be picked for inclusion in the maximal node matching (MNM) is edge (1, 3) that has three uncovered adjacent edges. As a result of this selection, the three adjacent edges of (1, 3) are said to be covered and removed from the graph. In the next iteration, we determine the number of uncovered adjacent edges for each of the remaining uncovered edges in the graph and select edge (2, 4) with three uncovered adjacent edges as the next edge for inclusion to the MNM. Finally, in

the third iteration, we have a tie between the three edges (5, 6), (5, 7) and (6, 7)—we break the tie arbitrarily by choosing edge (5, 6). The maximal node matching thus consists of the three edges $\{(1, 3), (2, 4), (5, 6)\}$ and their node weights are, respectively, $\{(0.5, 0.9), (0.7, 0.3), (0.1, 0.8)\}$. Unlike the MAM, we can see the difference in the node weights of the vertices in the MNM to be arbitrary (neither all low nor all high). The calculation for the assortative index is illustrated on the right side of Fig. 4. The assortative index (based on the node weights) of the MNM is -0.55 and the % of node matches is $6/7 = 86\%$. On the other hand, the assortative index of the MAM is 1.0 and the % of node matches is 57%. Thus, the toy example considered in Figs. 3 and 4 gives sufficient hints of the tradeoff between assortativity and maximal node matching (especially for sparse graphs) and this is further vindicated through the results presented and analyzed in Sects. 3–6.

Though the expected value for the assortative index of an MNM is 0 (to vindicate that the maximal node matching is independent of node weights); the assortativity index value of -0.55 observed for the graph in Fig. 4 is still far from -1 (an assortative index value of -1 would indicate the matching algorithm pairs nodes that are very dissimilar). Thus, the MNM for this toy example could still be considered somewhat neutral with respect to node weights. The maxi-

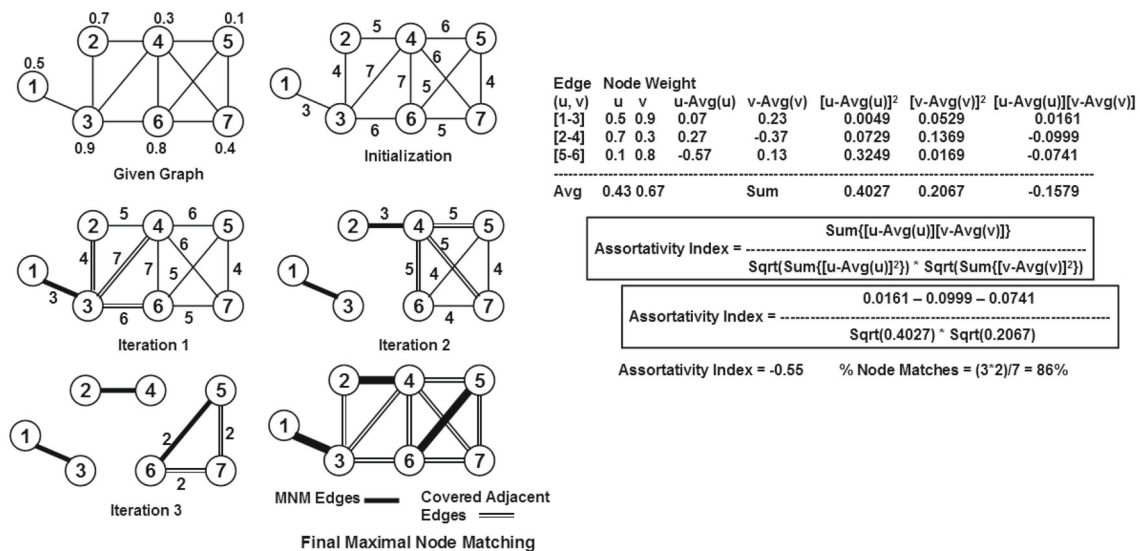


Fig. 4 Example to illustrate the execution of the algorithm for maximal node matching

mal node matching for the random graphs (generated based on the Erdos–Renyi model) with randomly assigned node weights (results analyzed in Sect. 5) incur an assortativity index close to 0 to vindicate that the maximal node matching is indeed independent of node weights.

3 Analysis of real-world network graphs

In this section, we present the results of the execution of the MAM and MNM algorithms on six real-world network graphs whose degree distribution ranges from Poisson (random networks) to Power-law (scale-free networks). The real-world networks are modeled as graphs with the nodes represented as vertices and links between any two nodes represented as edges (all edges are undirected). The implementation can work for any real-world network. The input to the implementation is an adjacency list of the real-world network wherein we store a list of edges (u, v) where $u < v$ (just as a convention we store the pair of vertices constituting an undirected edge in this format). A brief description of the six real-world networks available as .gml files at: <http://www-personal.umich.edu/~mejn/netdata/> is as follows: (i) US College Football Network [13] is a network of 115 football teams that played the Fall 2000 Football season in the US; each team is a node and there exists an edge between two nodes if and only if the corresponding teams have competed against each other in the earlier seasons. (ii) Dolphin Social Network [14] is a social network of 62 Dolphins living in the Doubtful Sound fjord of New Zealand; each Dolphin is modeled as a node and there exists an edge between two nodes if and only if the corresponding Dolphins are seen associated with each other. (iii) US Politics Books Network [15] is a net-

work of 105 books on US politics sold in amazon.com; each book is modeled as a node and there exists an edge between two nodes u and v if and only if customers who bought the book corresponding to node u also bought the book corresponding to node v and vice-versa. (iv) Zachary's Karate Club [16] is a network of 34 members of a Karate club at a US university in the 1970s; each member of the club is modeled as a node and there exists an edge between two nodes if and only if the corresponding members are friends. (v) Word Adjacencies Network [17] is a network of 112 words (adjectives and nouns) selected from the novel "David Copperfield" by Charles Dickens; each word is modeled as a node and there exists an edge between two words if and only if the two words have appeared adjacent to each other at least once in the book. (vi) US Airports 1997 Network [18] is a network of 332 airports; each airport is modeled as a node and there exists an edge between two nodes if and only if there is at least one direct flight connection between the corresponding airports. All the real-world networks are modeled as undirected graphs.

We characterize the nature of the degree distribution in these graphs on the basis of a metric, called the spectral radius ratio, for node degree [19]—defined as the ratio of the principal Eigenvalue (largest Eigenvalue) [8] of the adjacency matrix of the graph to that of the average node degree. The principal Eigenvalue of the adjacency matrix of a graph maximally captures the variation in the node degree. For networks that are completely random (i.e., there could exist an edge between any two vertices with a certain probability), the variation in the degree of the vertices in the graph is minimal and the spectral radius ratio for node degree is close to 1.0; on the other hand, for networks that exhibit a power-law degree distribution (majority of the nodes have low degree; but few

Table 1 Real-world networks and their degree distribution

#	Real-world network	# nodes	# edges	k_{\min}	k_{\max}	k_{avg}	λ_k
(i)	US College Football Network	115	613	7	12	10.66	1.01
(ii)	Dolphins' Social Network	62	159	1	12	5.13	1.40
(iii)	US Politics Books Network	105	441	2	25	8.40	1.41
(iv)	Karate Club Network	34	78	1	17	4.59	1.46
(v)	Word Adjacencies Network	112	425	1	49	7.59	1.73
(vi)	US Airports 1997 Network	332	2126	1	140	12.81	3.22

Table 2 Real-world networks and their analysis for maximal matching

#	Real-world network	Network A.index	MNM		MAM		Diff _{A.Index} = $MAM_{A.Index} - MNM_{A.Index}$
			% Node matches	A.Index	% Node matches	A.Index	
(i)	US College Football Net.	−0.04	99	0.51	95	0.81	0.30
(ii)	Dolphins' Social Net.	−0.04	93	−0.23	73	0.82	1.05
(iii)	US Politics Books Net.	−0.02	99	0.22	86	0.71	0.49
(iv)	Karate Club Net.	−0.48	76	−0.43	71	−0.13	0.30
(v)	Word Adjacencies Net.	−0.10	96	−0.18	78	0.50	0.68
(vi)	US Airports 1997 Net.	−0.21	83	−0.15	68	0.87	1.02

hub nodes have significantly larger degree), the variation in the degree of the vertices in the graph would be high and the spectral radius ratio for node degree would be far above 1.0. Table 1 lists the real-world networks analyzed in the increasing order of their spectral radius ratio for node degree (λ_k). We denote the minimum, maximum and average node degree for the graphs as k_{\min} , k_{\max} and k_{avg} , respectively.

As can be seen from Table 1, the US College Football Network exhibits a degree distribution that is very close to that of a random network (spectral radius ratio for node degree close to 1.0)—this is as expected, because other than the knock out games, each football team is more likely to play against all the other teams of the tournament in the round-robin games and thus, the number of football teams that each team has played against is quite close to the average number of teams that every team has played against. On the other hand, the US Airports network exhibits a scale-free distribution for node degree—indicating that there are few airports with degree as large as 140 (i.e., connections to 140 other airports in the network) while majority of the airports have fewer connections, leading to an average of 12.81 connections per airport. The other four real-world networks fall in between these two extremes.

We assume the node weights as node degree and calculate the assortativity index (hereafter, shortly referred to as A.Index) of the network (considering the set of all edges) and the assortativity index of the maximal matching obtained with the MAM and MNM algorithms. For all the six real-world networks, the A.Index values for each of the network graphs are negative, but majority of them are more close to 0

(indicating that the difference in the degrees of the end vertices of the edges is arbitrary and is neither too low nor too high). The A.Index values of the networks get more negative as the networks get increasingly scale-free, as is observed in the case of the Karate Club network, Word Adjacencies network and the US Airports network. We ran the MAM and MNM algorithms on each of the six real-world network graphs 100 times and averaged the results (presented in Table 2); this is to weed out any bias in the results due to the arbitrary breaking of ties among contending edges for inclusion to the set of edges constituting the maximal matching for both the MNM and MAM algorithms.

For each of the six real-world graphs, the MAM algorithm yielded a maximal matching that had a significantly larger A.Index compared to the matching obtained with the MNM algorithm. Neglecting the negative A.Index values obtained for the maximal matching to the Karate Club network under both the algorithms, the range of A.Index values obtained with the MAM algorithm across the other five real-world network graphs is 0.50–0.87, whereas the range of A.Index values obtained with the MNM algorithm across these real-world network graphs is −0.23 to 0.51. The median value for the A.Index of the maximal matching obtained with the MAM algorithm across the six real-world network graphs is 0.76, while the median value for the A.Index of the maximal matching obtained with the MNM algorithm is −0.17. As the A.Index values can range only from −1 to 1, a difference in the median A.Index values of $0.76 - (-0.17) = 0.95$ is very significant (as the difference in the A.Index values can be only

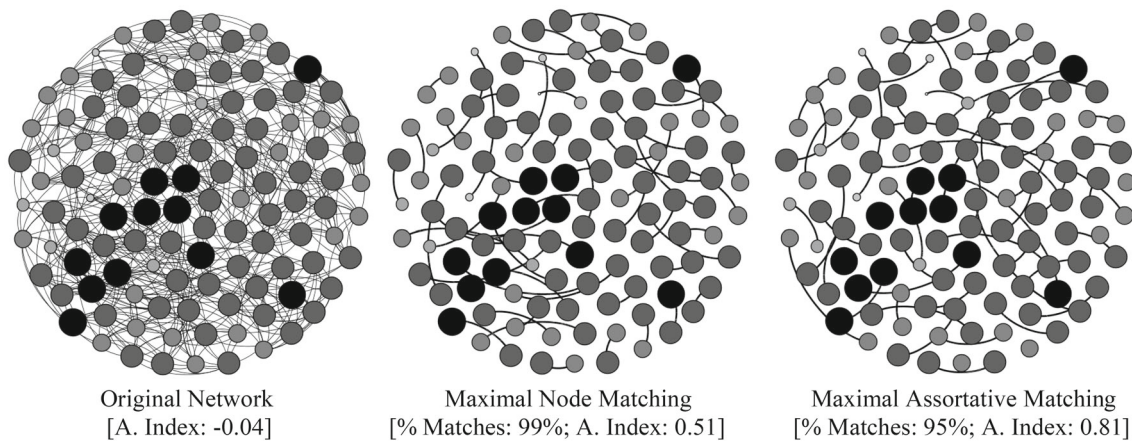


Fig. 5 US College Football Network: 115 nodes and 613 edges [spectral radius ratio: 1.01]

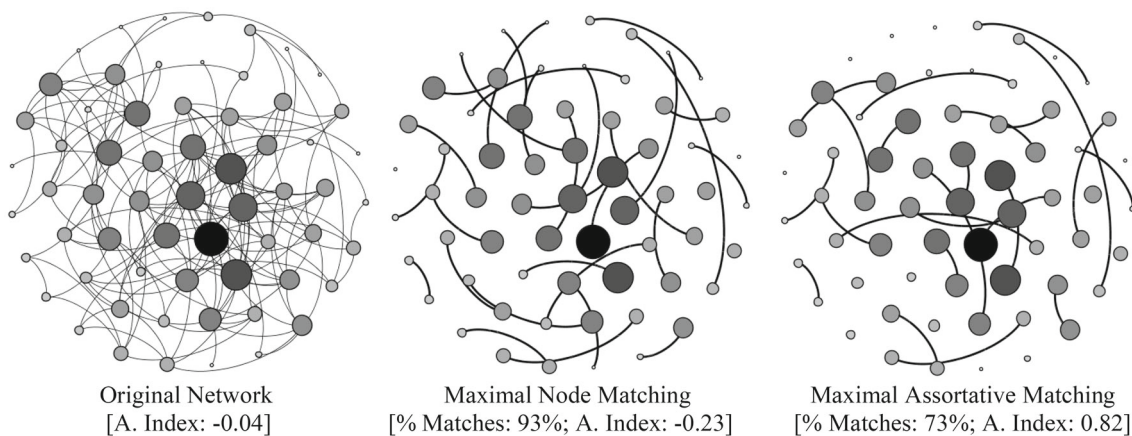


Fig. 6 Dolphins' Social Network: 62 nodes and 159 edges [spectral radius ratio: 1.40]

on a scale of 0–2). As a tradeoff for larger A.Index, we had expected the MAM algorithm to incur a relatively fewer node matches compared to that of the MNM algorithm. The results for the analysis of the real-world network graphs indicate that the tradeoff is indeed not very significant. We observe the % node matches for the MAM algorithm to range from 68 to 98 %, with a median of 75 %, whereas the % node matches for the MNM algorithm ranges from 76 to 99 %, with a median of 95 %. The difference in the median values for the % node matches is 20 % (on a scale of 0–100 %).

Figures 5, 6, 7, 8, 9 and 10 illustrate the maximal matching obtained for each of the real-world network graphs with respect to both maximal node matching and maximal assortative matching. The visualization is generated through Gephi [20]; the layout of these networks is based on the Fruchterman Reingold algorithm [21]. The larger and darker the node circles, the larger the degree for the node. As can be seen in the figures for all the six real-world network graphs, the maximal assortative matching (MAM) has a larger fraction of edges whose end vertices are more likely to have degrees close

enough to each other. On the other hand, we could observe some nodes remaining unpaired in the figures for the MAM, attributed to the relatively lower % of node matches.

The Gephi software package does not include algorithms for maximal matching. We primarily use Gephi for visualization; the implementations of the maximal matching algorithms proposed in this paper have been done from scratch through object-oriented programming in Java. The outputs of the algorithms (the edges chosen for maximal matching), saved as .csv files, are ported to Gephi through the “Data Laboratory” user interface for visualization. Gephi has a .jar toolkit at <https://gephi.org/toolkit/>. One could download this toolkit to their programming environment (say: NetBeans, <https://netbeans.org/>), transform the code for maximal matching algorithms to an API and integrate the API as a plug-in to Gephi.

The US Airports Network is the largest network (332 nodes and 2126 edges) we have analyzed in this paper. On a Dell Precision M4600 computer (Intel i7-2620M CPU @ 2.70 GHz; 8 GB RAM), the execution time of the MAM and MNM algorithms for the US Airports Network are, respec-

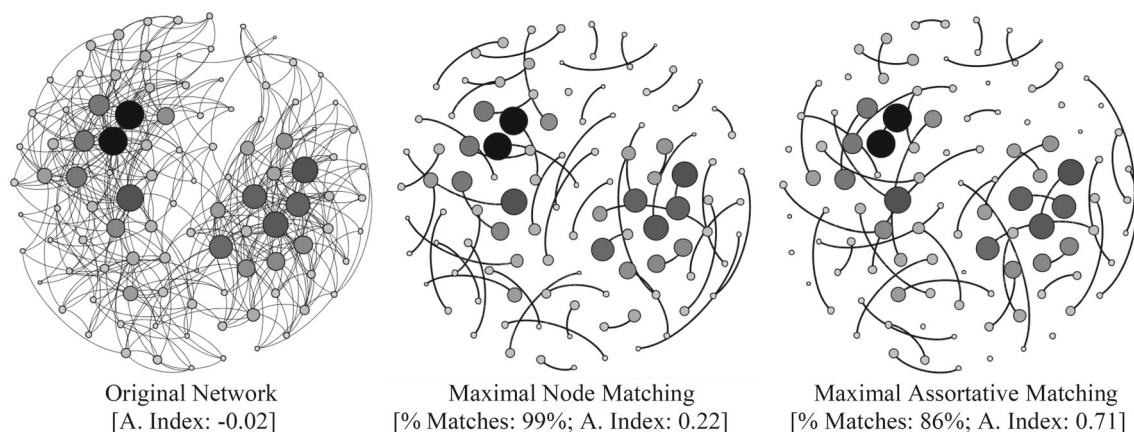


Fig. 7 US Politics Books Network: 105 nodes and 441 edges [spectral radius ratio: 1.41]

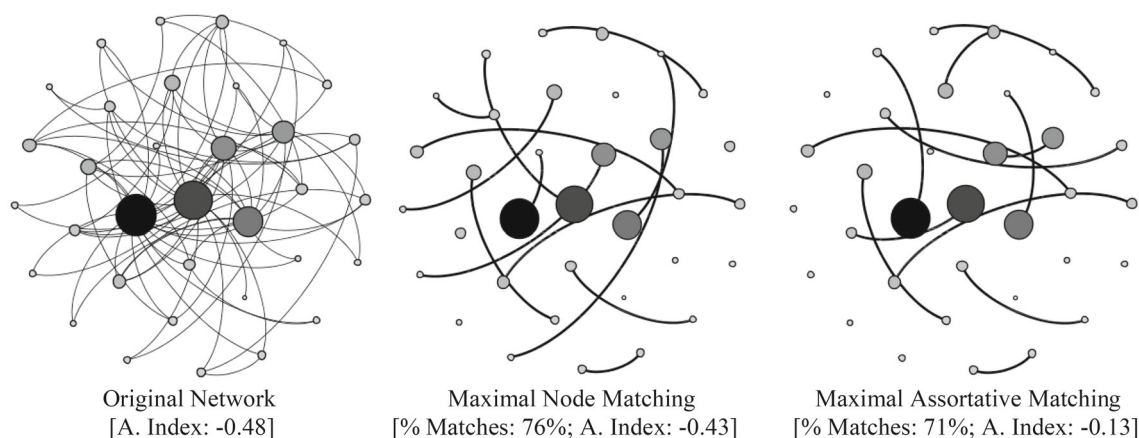


Fig. 8 Karate Club Network: 34 nodes and 78 edges [spectral radius ratio: 1.46]

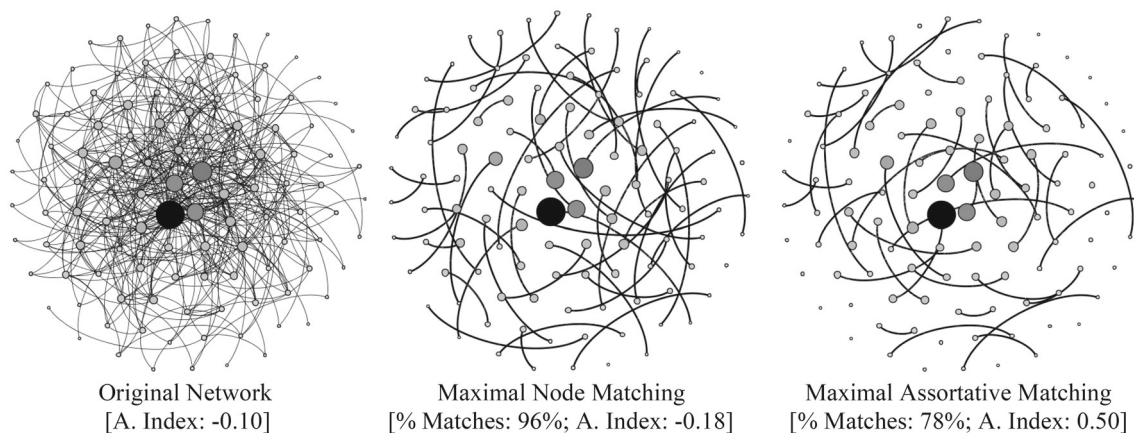


Fig. 9 Word Adjacency Network: 112 nodes and 425 edges [spectral radius ratio: 1.73]

tively, 0.23 and 0.19 ms. With a polynomial-time complexity $O(EV \log E)$ and actual execution times (in milliseconds, as observed above), we are confident that the proposed maximal matching algorithms are easily scalable for very large real-world networks of several hundreds and even thousands

of nodes. As mentioned earlier, the results of our maximal matching algorithms can be ported to any network visualization tool and the edges chosen for matching could be visualized; the resolution of the visualization is limited only to the tool being used.

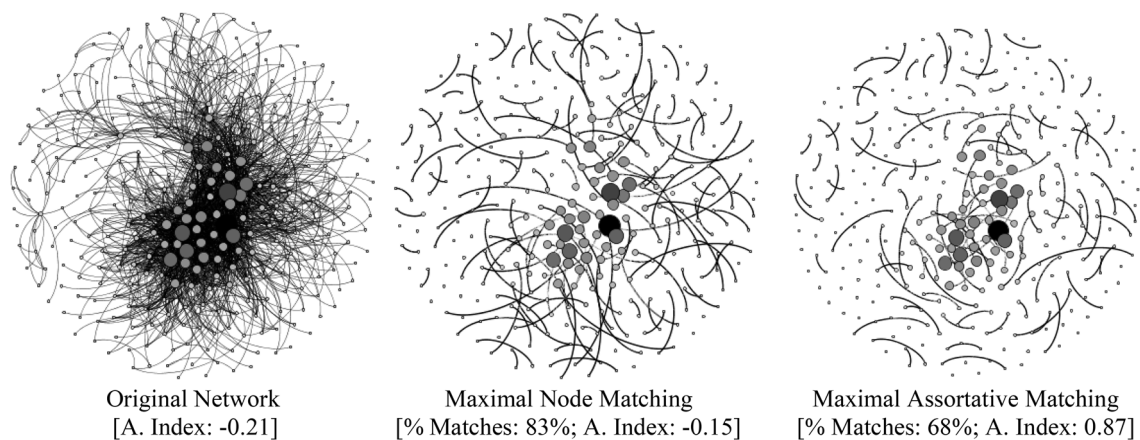


Fig. 10 US Airports (1997) Network: 332 nodes and 2126 edges [spectral radius ratio: 3.22]

Out of the results obtained for 100 trials, we pick the results of the trial that lie very close to that of the average values obtained for the assortativity of the network as well as the % node matches and the assortativity index for both the MAM and MNM. In this pursuit, we normalize the values for each of the above five metrics for each of the 100 trials as well as normalize the average values of the metrics for all the trials; for each trial, we determine the sum of the squares of the difference between the normalized values for the above five metrics and the normalized average values; we use the maximal assortative matching and maximal node matching of the trial that has the lowest error for the sum of the squares of the differences (as calculated above). The network graphs and the matching shown in Figs. 5, 6, 7, 8, 9 and 10 correspond to the graphs of the trials chosen as explained above.

With regard to the impact of the type of network on assortativity, we observe the random networks to incur larger A.index values for both MAM and MNM and this could be attributed to the relatively fewer number of adjacent edges per edge and the edges are evenly distributed across the entire network. On the other hand, for scale-free networks, a significant fraction of the edges are incident on the hubs and relatively lower fraction of the edges connect two non-hub nodes. As a result, the inclusion of an edge incident on a hub into a maximal matching leads to the coverage of a larger fraction of the uncovered edges as well as results in a low-degree node being paired with a high-degree node. Such scenarios are more evident in the case of maximal node matching, leading to much smaller negative A.Index values (reasonably lower than 0). The MAM algorithm is much more successful in identifying and including edges between two non-hub nodes (low-moderate degree nodes) as part of the maximal matching.

Compared to the other real-world network graphs, the relatively poor performance of both the maximal assortative

matching and maximal node matching for the Karate Club network could be attributed to the sparse and scale-free nature of the graph (34 nodes and 78 edges; the average node degree is 4.59 and the maximum node degree is 17); as a result, the inclusion of an edge in the MAM or MNM is more likely to result in the coverage of several other adjacent uncovered edges as well as result in the more likely pairing of a low-degree node with a high-degree node.

4 Analysis of random network graphs with node degree as node weights

In Sects. 4 and 5, we simulate the evolution of random network graphs generated using the well-known Erdos–Renyi model [10]. The model inputs two parameters: the total number of nodes (N) and the probability of a link (p_{link}) between any two nodes in the graph. As we simulate the evolution of an undirected random network, the links are bi-directional and we could assume that the end vertices of each link could be represented as an ordered pair (u, v) where u and v are the node IDs and $u < v$. We assume there are no self-loops and there is no more than one edge in the network. For an N node network, the maximum number of undirected links possible in the network is $N(N - 1)/2$. We consider every such possible link in the network and generate a random number to decide whether to include the link in the network or not. If the random number generated for a pair (u, v) is less than or equal to p_{link} , then we include the link (u, v) in the network; otherwise, not. As it is obvious, the larger the value of p_{link} , the larger the number of links in the random network graph as well as larger the chances for the network to have a degree distribution wherein the degree of each node is closer to the average node degree.

The total number of nodes considered in the simulations for this section is $N = 100$ nodes. The values used for the probability of link between any two nodes in the network

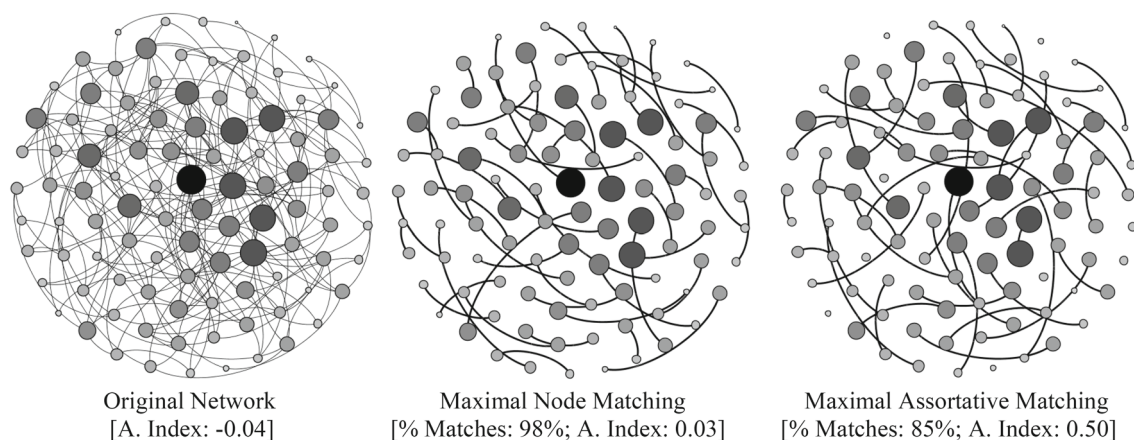


Fig. 11 Random network: 100 nodes; probability of link between any two nodes—0.05

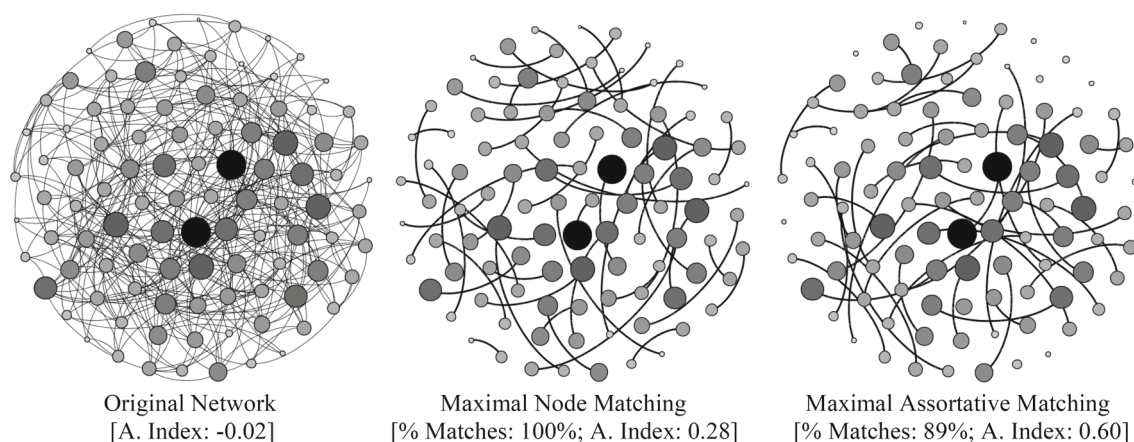


Fig. 12 Random network: 100 nodes; probability of link between any two nodes—0.07

(p_{link}) are 0.05, 0.07, 0.10, 0.15, 0.20, 0.30, 0.40 and 0.50. For each p_{link} value, we run 100 trials of the network evolution and analyze the assortativity of the network as well as evaluate the % of node matches and assortativity of the maximal matching obtained with both the MAM and MNM algorithms. In this section, node degree is used as node weight for the assortativity calculations.

As explained in Sect. 3, we pick the trial network whose values for the above five metrics lie close to the average values for these metrics across all the 100 trials (according to the sum of the squares of the differences between the normalized values of the metrics for the individual trials and the overall normalized average values) and represent in Figs. 11, 12, 13, 14, 15, 16, 17 and 18 the network graph and maximal matching (with respect to both MAM and MNM) obtained for the network graph of the chosen trial.

We observe that the random networks for all the 100 trials generated with $p_{\text{link}} \geq 0.05$ to be connected. Even though the number of links in the network increases with increasing p_{link} values, the assortativity of the set of all edges in a ran-

dom network remains close to 0 for all the p_{link} values. This vindicates the random nature of the distribution of the edges among the vertices as per the Erdos–Renyi model.

With regard to the % of node matches, we start observing a 100 % node match with the MNM algorithm with p_{link} values of 0.07 or above, whereas the % of node matches with the MAM algorithm is 85 % for p_{link} value of 0.05 and reaches 99 % for p_{link} value of 0.5; the % of node matches for MAM crosses 95 % when p_{link} is 0.15. However, the tradeoff is quite high with respect to the assortativity index (A.Index). The A.Index of the maximal node matching is significantly low compared to that of the maximal assortative matching. The A.Index of MNM and MAM are, respectively, 0.03 and 0.5 when p_{link} is 0.05 and reaches 0.60 and 0.84 when p_{link} value is 0.3. The A.Index does not increase appreciably for both the MAM and MNM (especially for the MNM) as we further increase the p_{link} value. The average A.Index values observed for the MNM and MAM are, respectively, 0.64 and 0.90 when the p_{link} value is 0.90. This is a significant observation that has been hitherto not reported in the literature for random

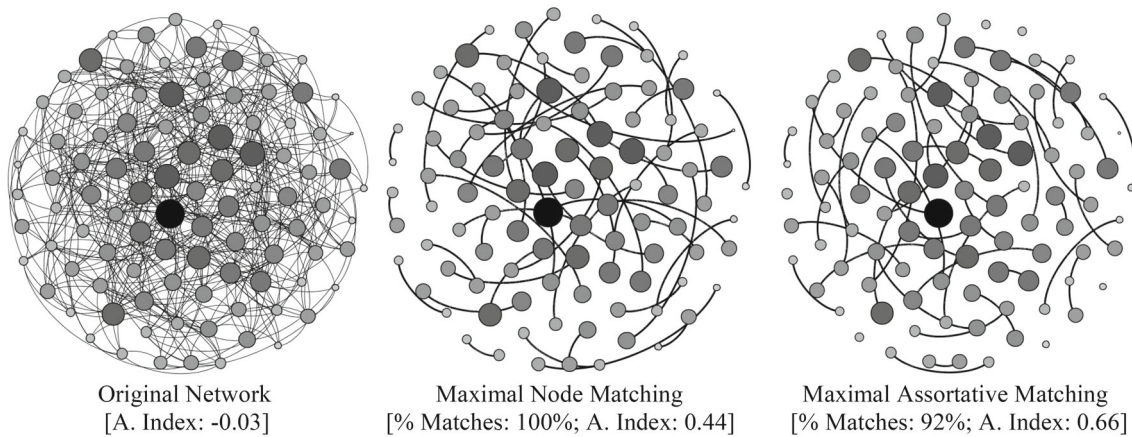


Fig. 13 Random network: 100 nodes; probability of link between any two nodes—0.10

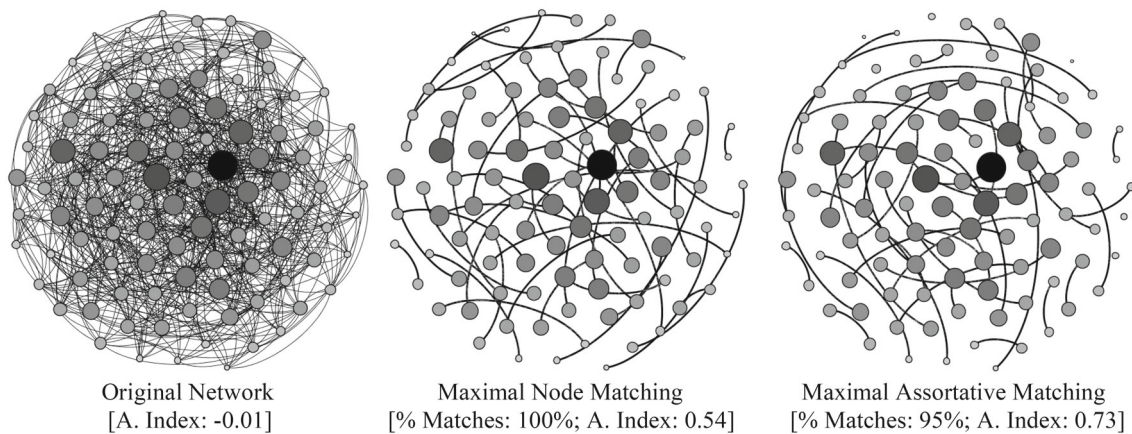


Fig. 14 Random network: 100 nodes; probability of link between any two nodes—0.15

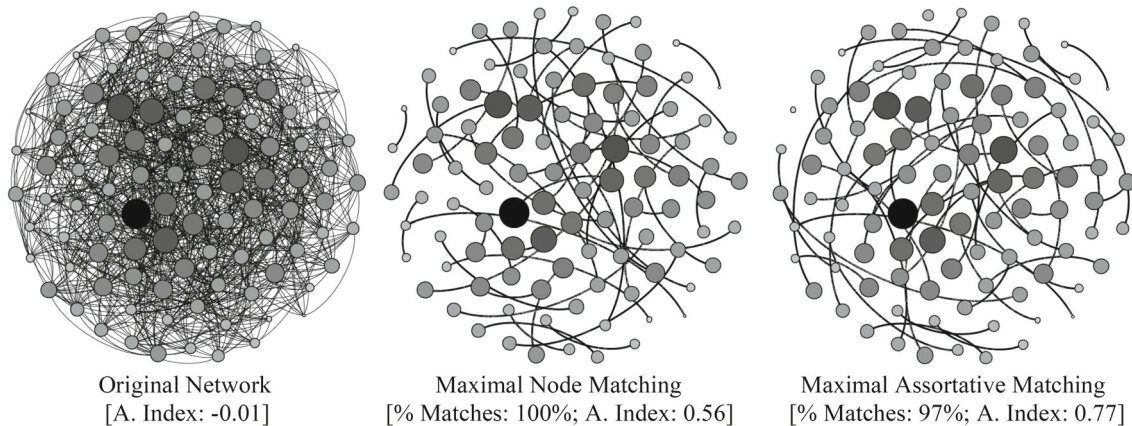


Fig. 15 Random network: 100 nodes; probability of link between any two nodes—0.20

networks. Figure 19 illustrates the nature of increase in % of node matches and the assortativity index values as we increase the p_{link} values from 0.05 to 0.50 as explained above. The values reported in Fig. 19 are the average values obtained from the 100 trial runs for each p_{link} value.

5 Analysis of random network graphs with random node weights

In this section, we present the results for the percentage of node matches and assortativity index incurred with the max-

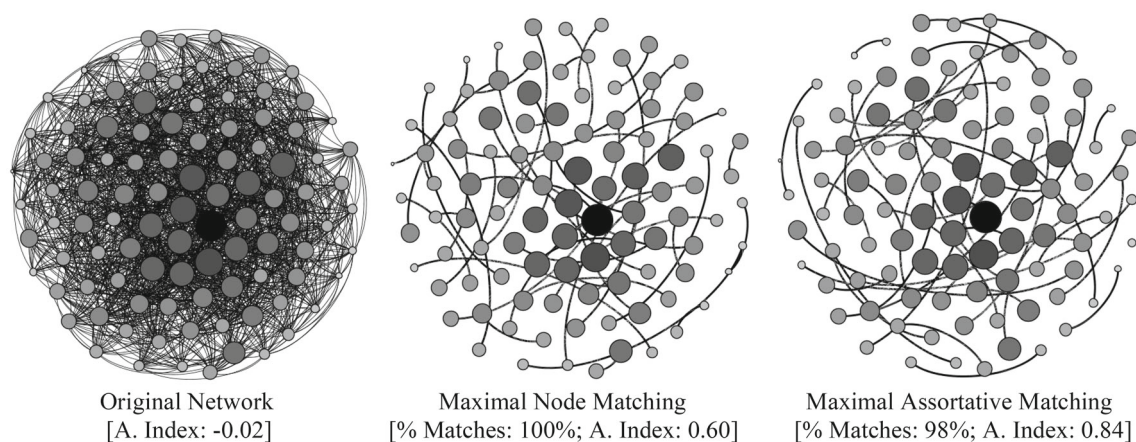


Fig. 16 Random network: 100 nodes; probability of link between any two nodes—0.30

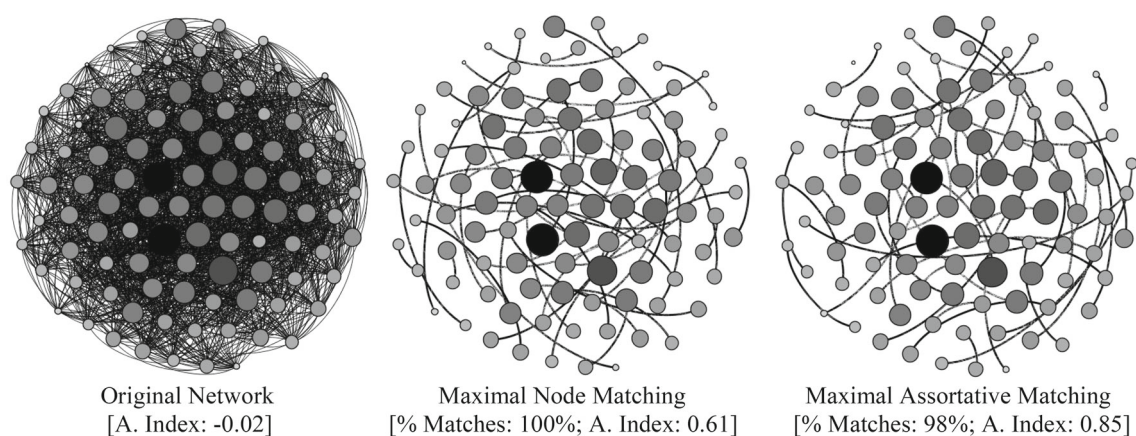


Fig. 17 Random network: 100 nodes; probability of link between any two nodes—0.40

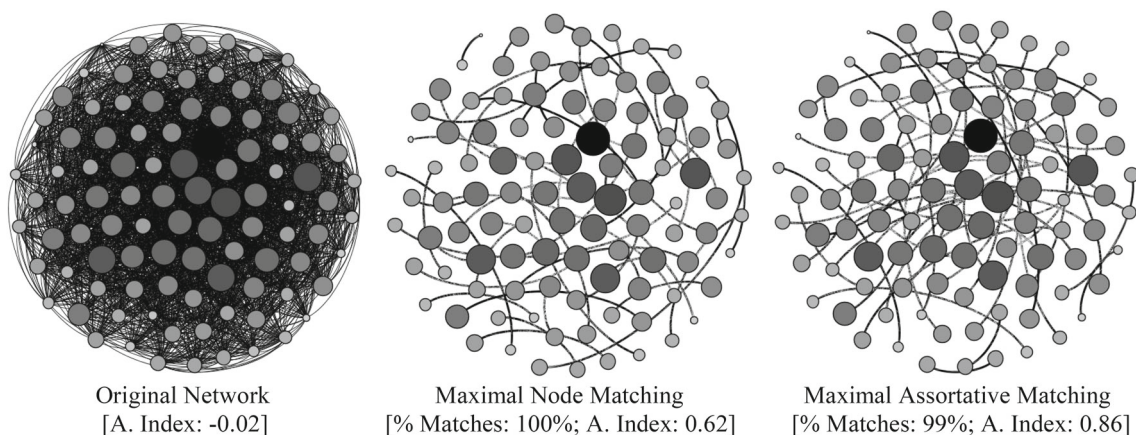


Fig. 18 Random network: 100 nodes; probability of link between any two nodes—0.50

imal node matching and maximal assortative matching for random networks generated under the Erdos–Renyi model wherein the node weights are random numbers generated from 0 to 1. We conducted the simulations with 100 trials for each p_{link} value and averaged the results for the

network assortativity as well as the % of node matches and assortativity index for both the maximal node matching and maximal assortative matching. The results presented in Fig. 20 indicate the average values for these metrics from the 100 trials. Also, the network graphs presented in Figs.

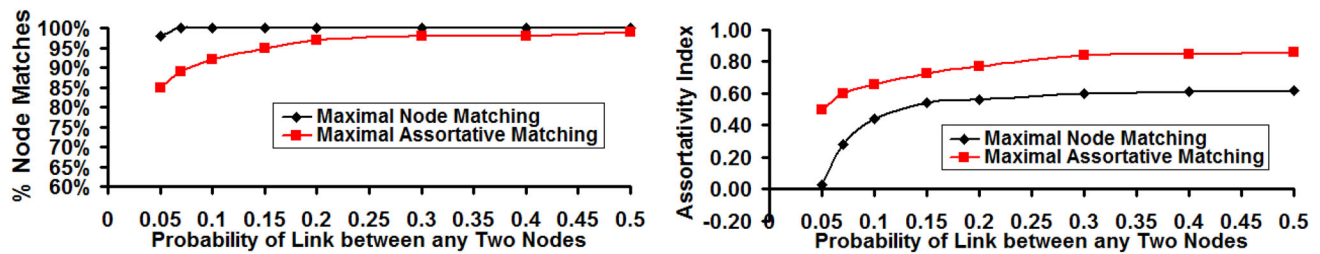


Fig. 19 Random networks with node degree as node weights: distribution of the percentage of node matches and assortativity index as a function of the probability of link between any two nodes

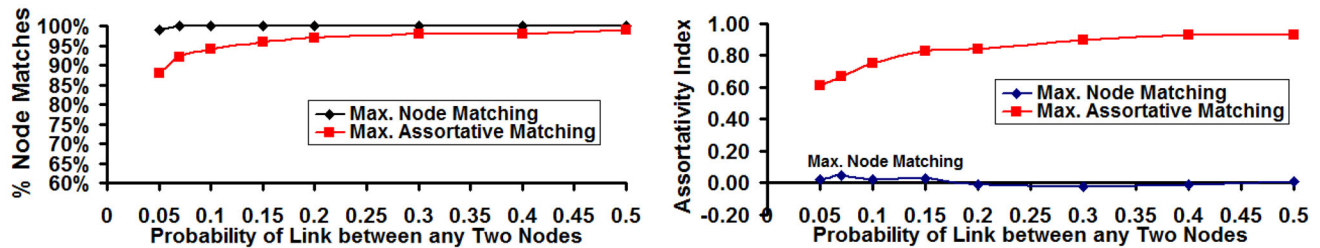


Fig. 20 Random networks with random node weights: distribution of the percentage of node matches and assortativity index as a function of the probability of link between any two nodes

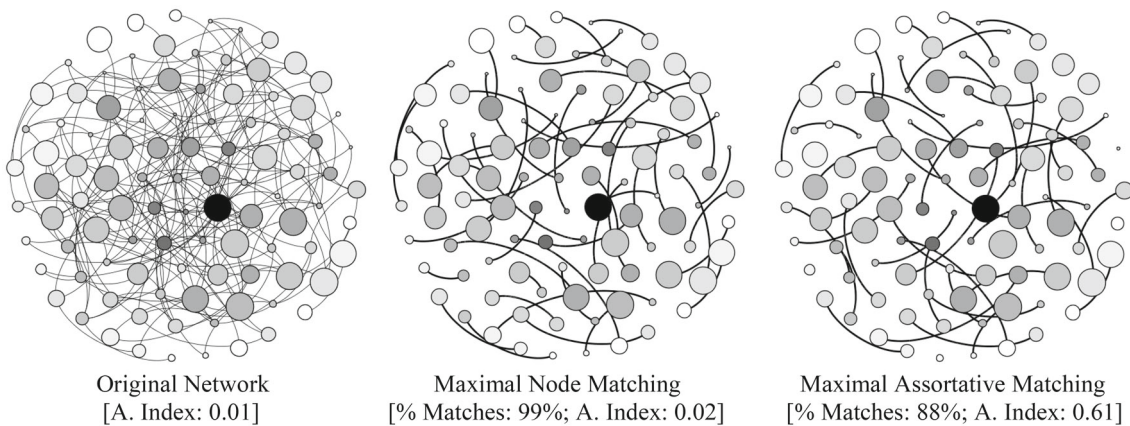


Fig. 21 Random network: 100 nodes; probability of link between any two nodes—0.05

21, 22, 23, 24, 25, 26, 27 and 28 are representative network graphs whose values for the above five metrics are close to that of the average values observed for these metrics in the 100 trials such that the sum of the squares of the difference in the normalized values for these metrics is the minimum. In Figs. 21, 22, 23, 24, 25, 26, 27 and 28, the size of the node circles is a measure of the node weight (the larger the node weight, the larger is the size of a node and vice-versa); on the other hand, the darkness of a node circle is a measure of the node degree (the more darker—black—a node is, the larger is its node degree).

As explained in Sects. 1 and 2, the maximal node matching is independent of node weights; as a result, we expect the assortativity index of maximal node matching to be close

to 0 for all values of p_{link} and it is confirmed through the simulations. On the other hand, though it was not obvious before the simulations, for a given p_{link} value, we observe the assortativity index of the maximal assortative matching (with random node weights) to be slightly higher (the difference is as large as 0.1 in a scale of 0–2) than the assortativity index of the maximal assortative matching with node degree as node weights. Though the difference in the assortativity index values for maximal assortative matching with the above two categories of node weights could be observed for all p_{link} values, the difference is relatively more prominent for random networks with lower p_{link} values and reduces as the p_{link} value increases. As can be observed from Fig. 20, the curve for the assortativity index for maximal assortative matching with random node weights becomes flat starting

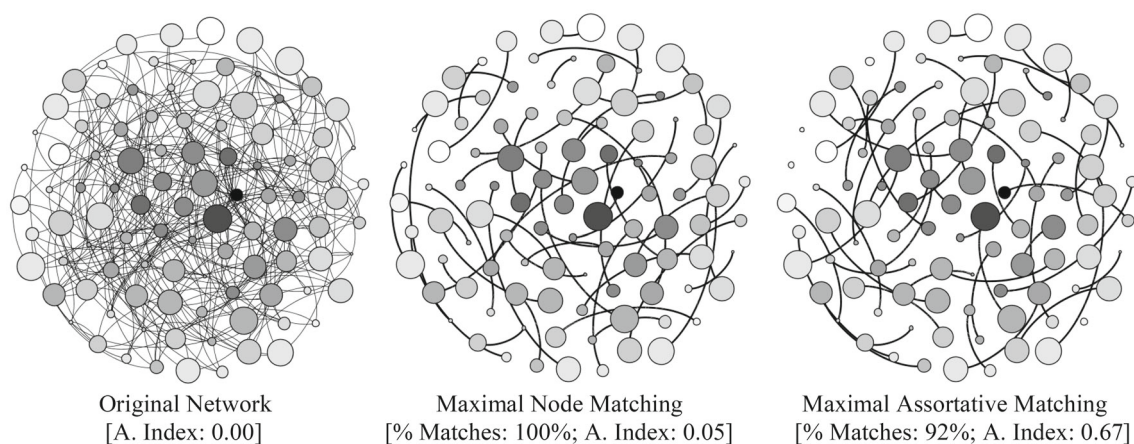


Fig. 22 Random network: 100 nodes; probability of link between any two nodes—0.07

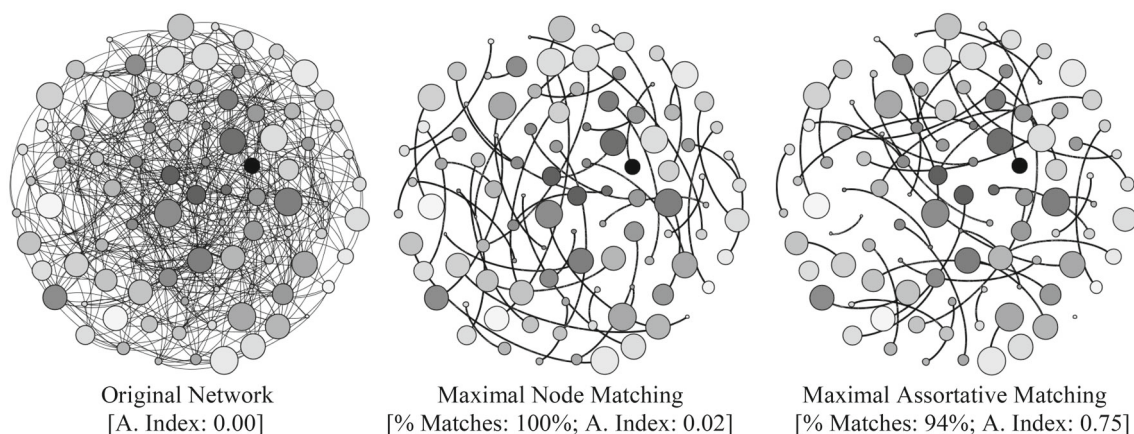


Fig. 23 Random network: 100 nodes; probability of link between any two nodes—0.10

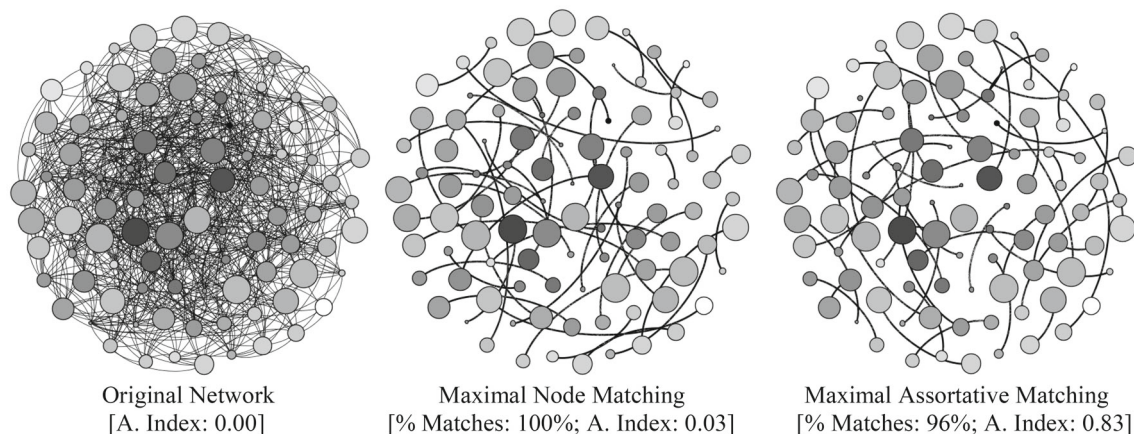


Fig. 24 Random network: 100 nodes; probability of link between any two nodes—0.15

from p_{link} value of 0.40 (the assortativity index curve for the MAM with node degree as node weights became flat starting from p_{link} value of 0.30).

An interesting observation is that (in addition to incurring a relatively larger assortativity index) the % of node

matches obtained with the MAM algorithm for random network graphs with random node weights is even slightly larger than the % of node matches obtained with the MAM algorithm for random network graphs with node degree as node weights, especially for networks formed with lower

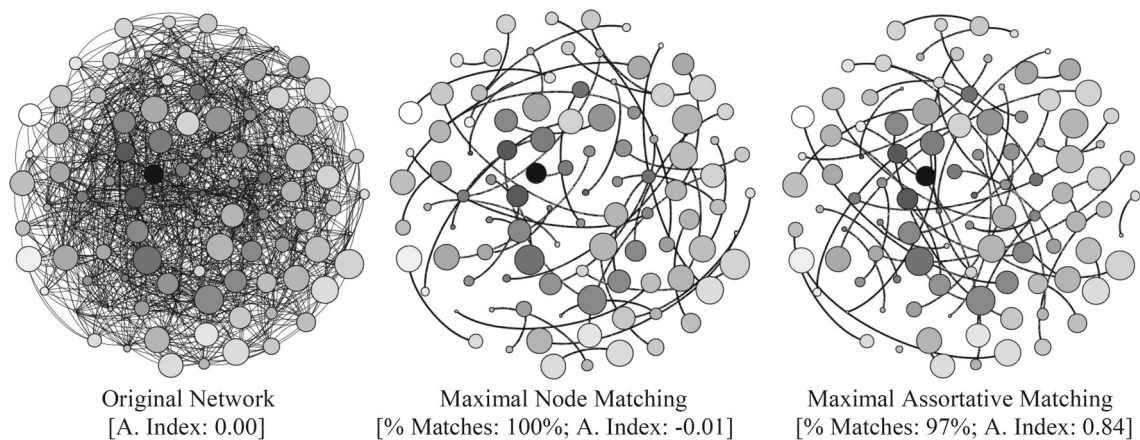


Fig. 25 Random network: 100 nodes; probability of link between any two nodes—0.20

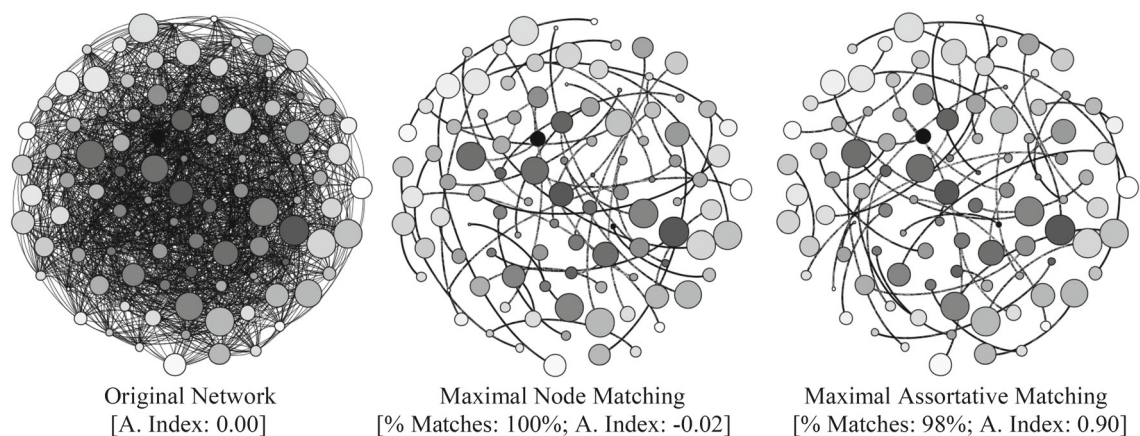


Fig. 26 Random network: 100 nodes; probability of link between any two nodes—0.30

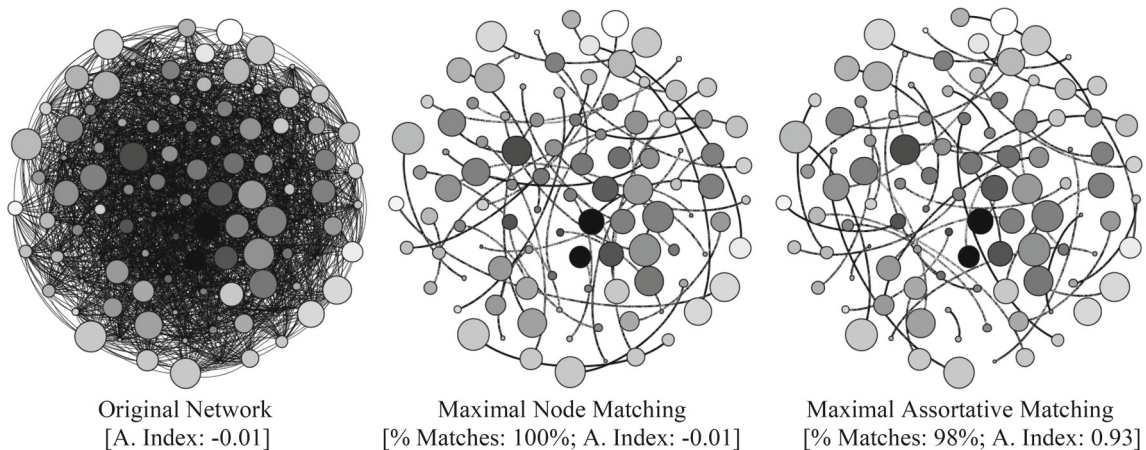


Fig. 27 Random network: 100 nodes; probability of link between any two nodes—0.40

p_{link} values. Overall, the maximal assortative matching algorithm could give even relatively better optimal results (with respect to both assortativity index and % of node matches) for random network graphs with random node weights and the tradeoff in the values incurred for the above two metrics is relatively less pronounced than what

is observed in random network graphs with node degree as node weights.

As we expect node weights in social networks to be not only a measure of the node degree, the MAM algorithm could be very useful to match vertices with any measure of node weights, especially in social network graphs that are not very

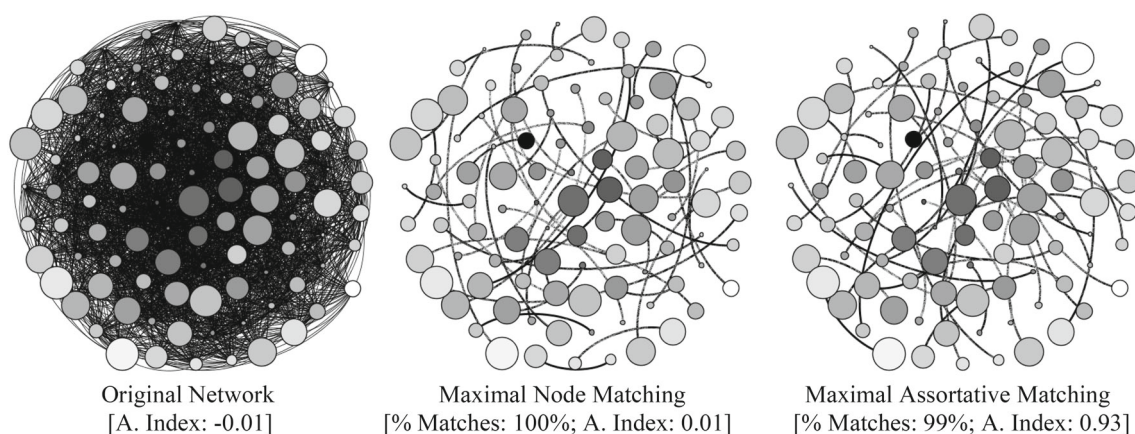


Fig. 28 Random network: 100 nodes; probability of link between any two nodes—0.50

dense. This vindicates the wider scope of application of the proposed maximal assortative matching (MAM) algorithm; the algorithm could give even better optimal results (with respect to assortativity) for random graphs with node weights that are independent of node degree.

6 Analysis of scale-free network graphs with node degree as node weights

In this section, we present the results of the execution of the maximal assortative matching and maximal node matching algorithms on scale-free network graphs that are generated with the well-known Barabasi–Albert (BA) model [11]. The evolution of a scale-free network under the BA model is explained briefly below: We start with an initial number of nodes (n_{init}) and setup links between them in such a way that each node has at least one link. The node IDs are assigned as $1, 2, \dots, n_{\text{init}}$. After this initialization, we start a timer ($t = n_{\text{init}} + 1, n_{\text{init}} + 2, \dots, n_{\text{total}}$) introducing new nodes to the network, one at a time (with IDs corresponding to the time of introduction of the node). We setup $\text{links}_{\text{new}}$ links to a newly introduced node (connecting it to the existing nodes in the network; not more than one link per node). If $\text{links}_{\text{new}}$ is greater than or equal to the total number of nodes existing in the network at the time of introduction of a node, then the newly introduced node is simply connected to each of the existing nodes (one link per node). If $\text{links}_{\text{new}}$ is less than the total number of nodes existing in the network at the time of introduction of a node, then the existing nodes are chosen for a link probabilistically according to the formulation explained below. The idea is to give preference for nodes that have a relatively higher degree (i.e., the BA model follows the rich-gets-richer preferential attachment phenomenon).

The probability for an existing node i to be chosen to have a link with a newly introduced node j is proportional to the

degree of the node i at the time of introduction of node j (since node i has been already introduced to the network at the time of introduction of node j , going by the above convention, $j > i$ and the IDs of all the existing nodes in the network will be $1, 2, \dots, j-1$). Let t_j denote the time of introduction of node j . Let $k_i^{t_j-1}$ be the degree of node i just before the introduction of node j to the network (i.e., at the end of the time of introduction of node $j-1$). Before any new link is added due to the introduction of node j , we compute the unnormalized probability $P_i^{j, \text{unnorm}} = \frac{k_i^{t_j-1}}{\sum_{id=1}^{j-1} k_{id}^{t_j-1}}$ with which an existing node i gets a link.

To decide which of the $1, 2, \dots, j-1$ nodes get the first link to node j , we divide the range $(0, \dots, 1]$ proportionally among the $j-1$ nodes such that node 1 gets the sub range $(0, \dots, P_1^{j, \text{unnorm}}]$, node 2 gets the sub range $(P_1^{j, \text{unnorm}}, \dots, P_2^{j, \text{unnorm}}]$, etc., and node $j-1$ gets the sub range $(P_{j-1}^{j, \text{unnorm}}, \dots, 1]$. We generate a random number in the range $(0, \dots, 1]$ and depending on which sub range the random number falls into, the corresponding node is selected to have the first link to the newly introduced node j ; the chosen node is not considered for the inclusion of any other new link (among the $\text{links}_{\text{new}}$ links) to be added during the introduction of node j . Let $\text{Neighbors}(j)$ be the set of nodes that have already had a link with the newly introduced node j . To decide which of the $\{1, 2, \dots, j-1\}$ - $\text{Neighbors}(j)$ candidate nodes get a link with node j , we normalize the unnormalized probability of the candidate nodes $i \in \{1, 2, \dots, j-1\}$ - $\text{Neighbors}(j)$ as follows: $\forall i \in \{1, 2, \dots, j-1\}$ - $\text{Neighbors}(j)$, $P_i^{j, \text{norm}} = \frac{P_i^{j, \text{unnorm}}}{\sum_{id \in \{1, 2, \dots, j-1\}$ - $\text{Neighbors}(j)} P_{id}^{j, \text{unnorm}}}$. We divide the range $(0, \dots, 1]$ proportionally among the candidate nodes $i \in \{1, 2, \dots, j-1\}$ - $\text{Neighbors}(j)$ according to the $P_i^{j, \text{norm}}$ values, similar to what was explained for the introduction of the first link. We generate a random number in the range

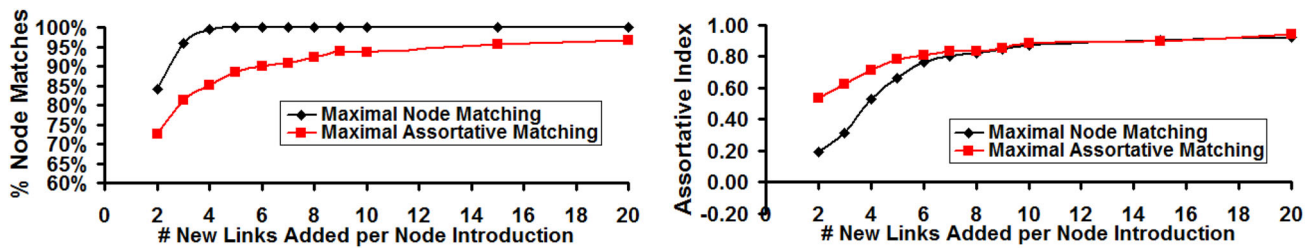


Fig. 29 Scale-free networks: average values for % of node matches and assortativity index (initial # nodes: 3; total nodes: 100)

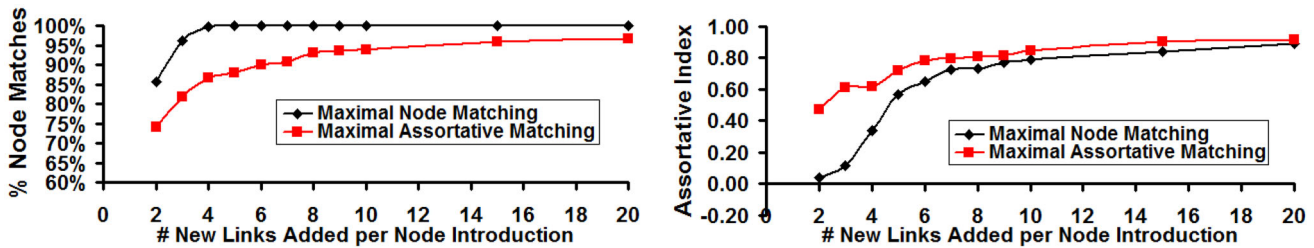


Fig. 30 Scale-free networks: average values for % of node matches and assortativity index (initial # nodes: 10; total nodes: 100)

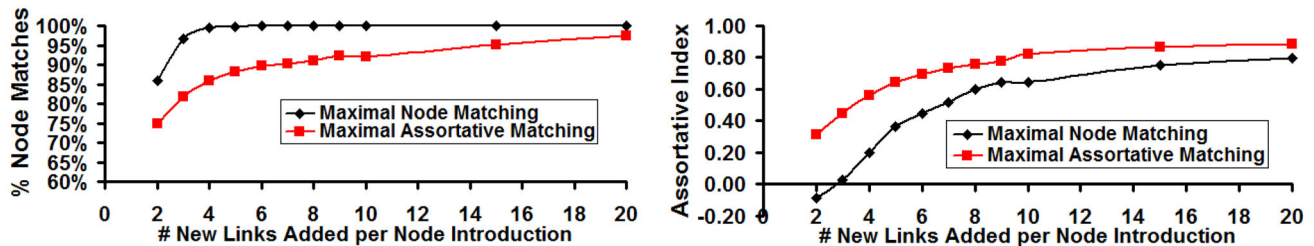


Fig. 31 Scale-free networks: average values for % of node matches and assortativity index (initial # nodes: 20; total nodes: 100)

(0, ..., 1] and whichever candidate node falls in the normalized range of probabilities, that node gets the new link. We repeat the above procedure until all of the $links_{new}$ links are added to a newly introduced node j .

To conduct the assortativity analysis, we simulate the evolution of a scale-free network under the above explained BA model with a total of 100 nodes (n_{total}): varied the initial number of nodes (n_{init}) with values of 3, 10 and 20, and varied the initial number of links per node at the time of its introduction ($links_{new}$) with values of 2, 3, 4, 5, 6, 7, 8, 9, 10, 15 and 20. We ran 100 trials of the simulations for each combination of n_{init} and $links_{new}$ values and averaged the results for network assortativity, % of node matches and assortativity index (A.Index) with respect to both maximal node matching (MNM) and maximal assortativity matching (MAM). Figures 29, 30 and 31 illustrate the averaged values for the % of node matches and A.Index for each combination of n_{init} and $links_{new}$ values listed above. Figures 32, 33, 34, 35, 36, 37, 38, 39 and 40 illustrate the scale-free networks evolved using n_{init} values of 3, 10 and 20 and $links_{new}$ values of 2, 5 and 20. Node degree is used as node weight in the assortativity calculations. The larger and darker is the circle

for a node in Figs. 32, 33, 34, 35, 36, 37, 38, 39 and 40, the larger is its degree and vice-versa.

Some interesting observations can be made from the results presented in Figs. 29 and 31. For a given number of new links added per node introduction, the % of node matches does not appreciably change for both the MAM and MNM as we increase the initial number of nodes from 3 to 10 and further to 20. On the other hand, for a given number of new links added per node introduction, the assortativity index for both the MNM and MAM decreases significantly as we increase the initial number of nodes from 3 to 10 and further to 20 (especially, for lower values of the number of new links added per node introduction). This could be attributed to the relatively sparse nature of the scale-free networks and a larger variation in node degree (refer to Figs. 32, 33, 34, 35, 36, 37, 38, 39, 40) as we increase the initial number of nodes (for a fixed value of the initial number of links added per node introduction), especially for lower values of the number of new links added per node introduction.

The initial number of nodes setup during the evolution of the scale-free network form the core of the network to which the newly introduced nodes get attached to. As a result,

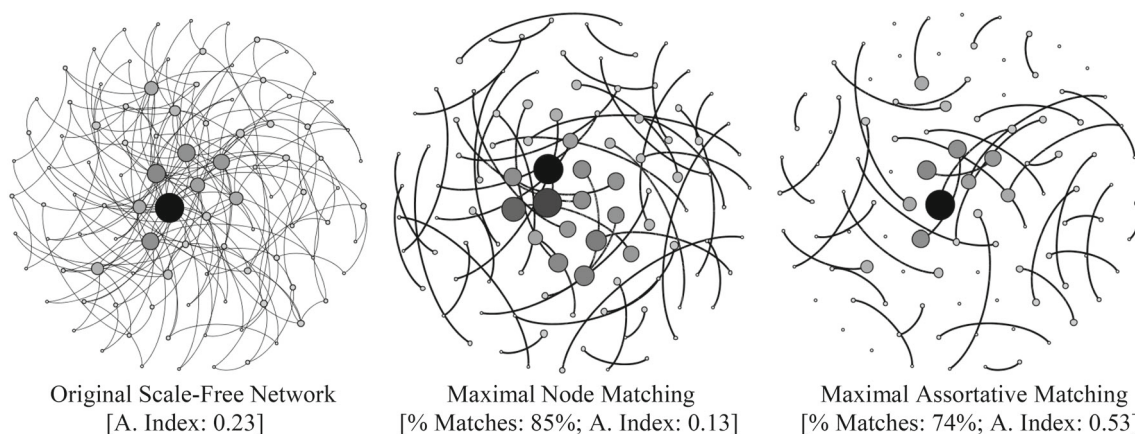


Fig. 32 Scale-free network (100 nodes)—BA model: initial—3 nodes; 2 links per node addition [spectral radius ratio for node degree: 1.78]

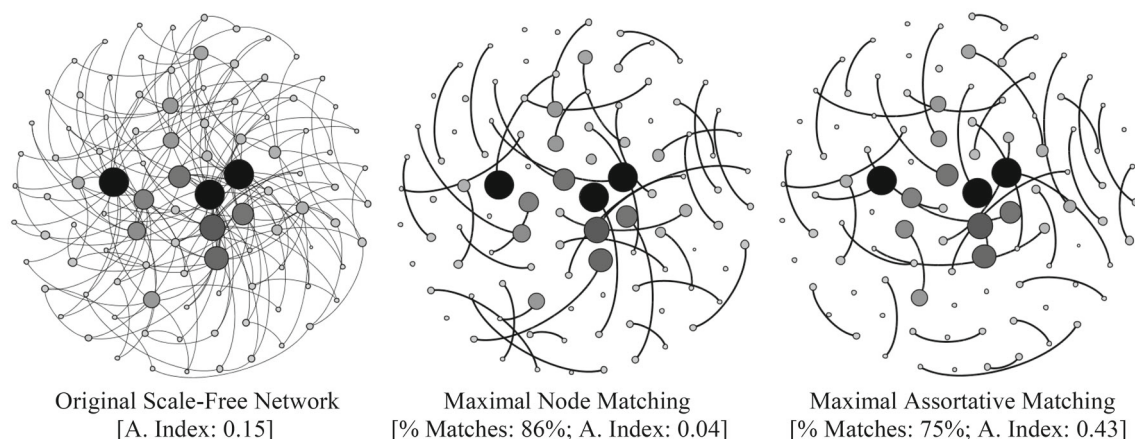


Fig. 33 Scale-free network (100 nodes)—BA model: initial—10 nodes; 2 links added per node introduction [spectral radius ratio for node degree: 1.62]

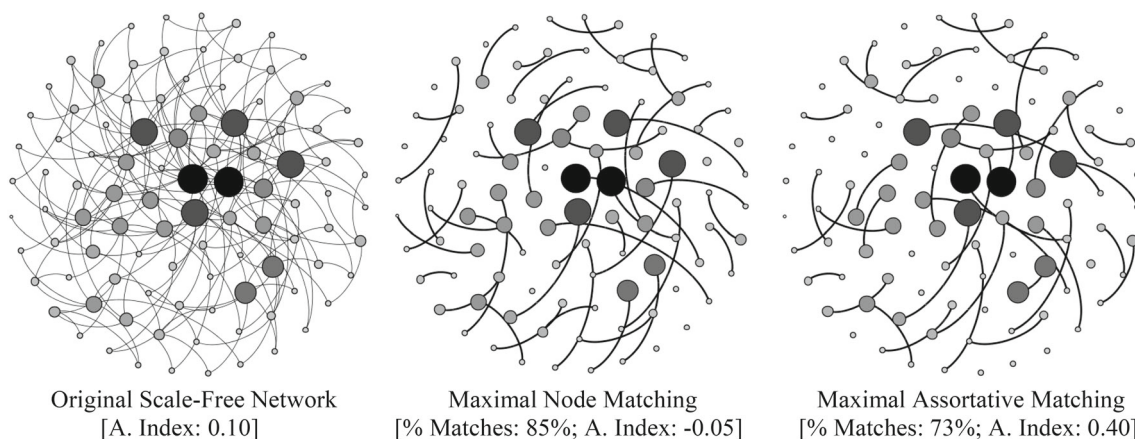


Fig. 34 Scale-free network (100 nodes)—BA model: initial—20 nodes; 2 links added per node introduction [spectral radius ratio for node degree: 1.49]

the initial set of nodes are bound to have a considerably larger degree than the newly introduced nodes (especially for smaller values of new links added per node introduction). If the initial number of nodes is high and the number of new

links added per node is low, the network is more sparse and also relatively more scale-free (vindicated by larger values for the spectral radius ratio for node degree, as in Figs. 32, 33, 34): with a concentration of only few hubs, most of the

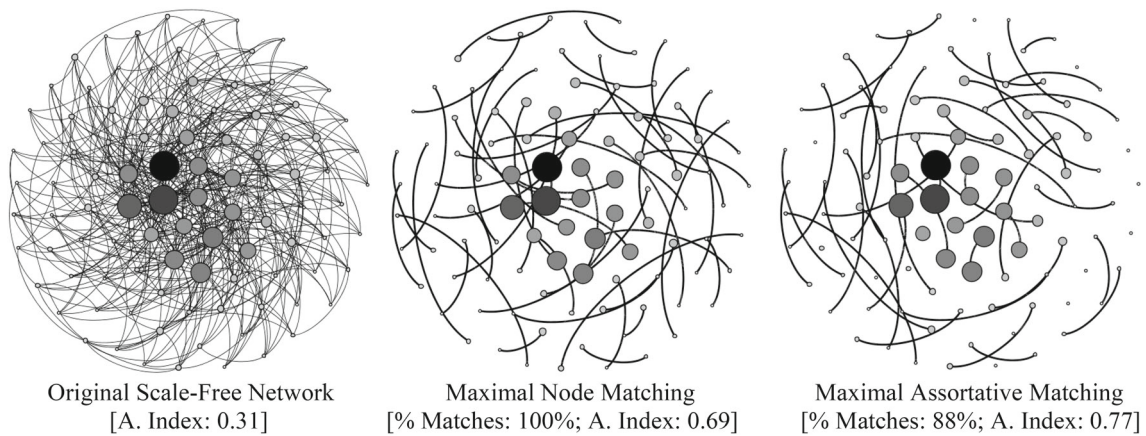


Fig. 35 Scale-free network (100 nodes)—BA model: initial—3 nodes; 5 links added per node introduction [spectral radius ratio for node degree: 1.46]

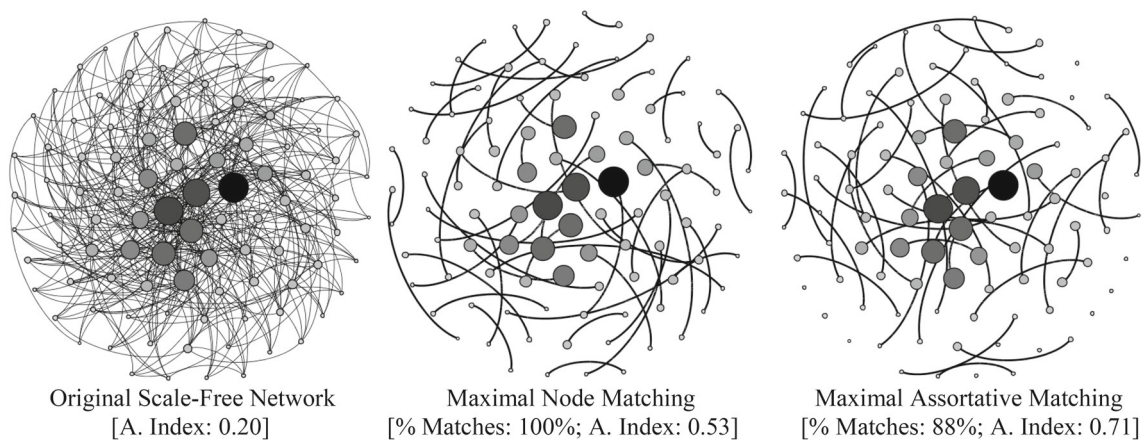


Fig. 36 Scale-free network (100 nodes)—BA model: initial—10 nodes; 5 links added per node introduction [spectral radius ratio for node degree: 1.35]

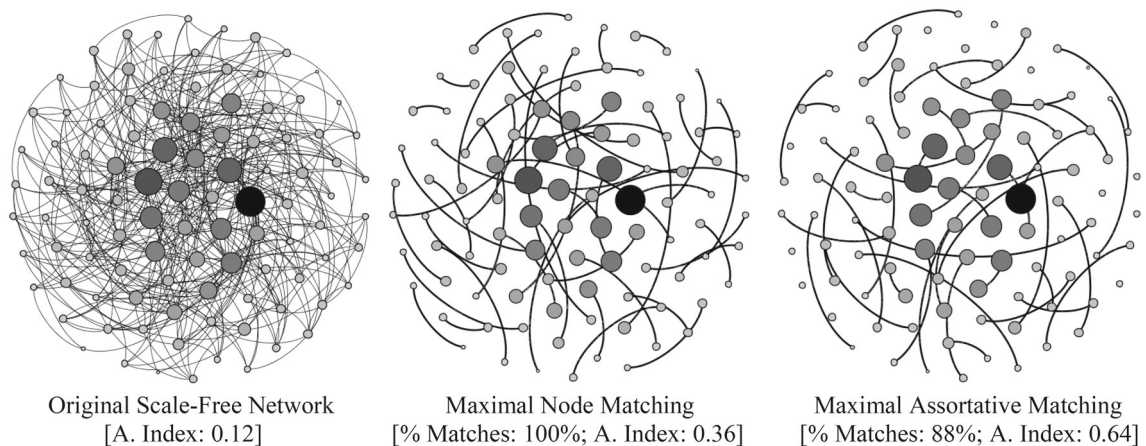


Fig. 37 Scale-free network (100 nodes)—BA model: initial—20 nodes; 5 links added per node introduction [spectral radius ratio for node degree: 1.29]

links are links involving a low-degree node connected to a high-degree node (results in a lower assortativity index for the matching algorithms).

As we increase the initial number of nodes and/or the number of new links added per node introduction, the number of high-degree nodes increases and the variation in the degree

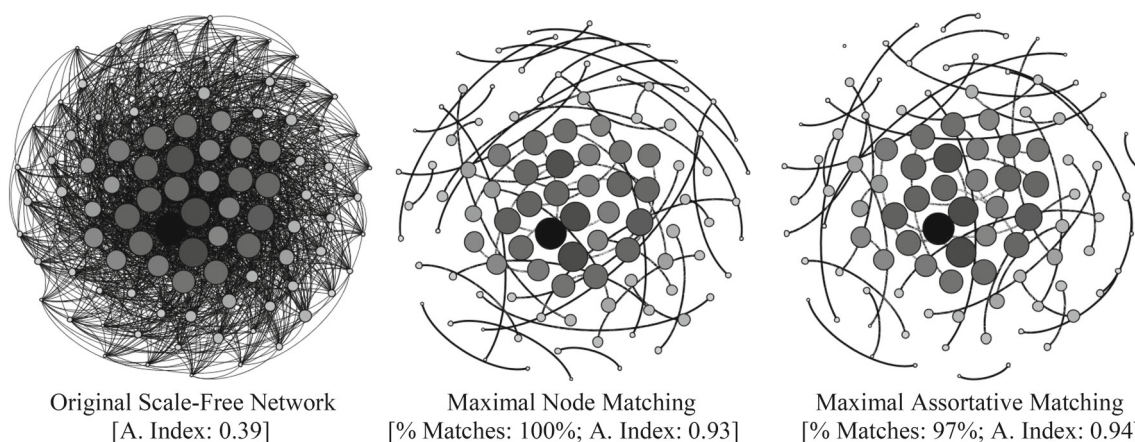


Fig. 38 Scale-free network (100 nodes)—BA model: initial—3 nodes; 20 links added per node introduction [spectral radius ratio for node degree: 1.16]

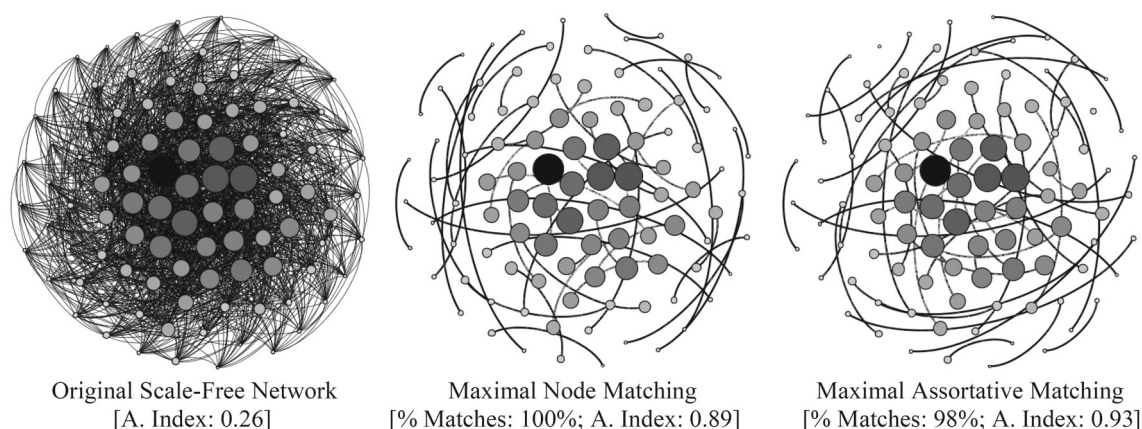


Fig. 39 Scale-free network (100 nodes)—BA model: initial—10 nodes; 20 links added per node introduction [spectral radius ratio for node degree: 1.14]

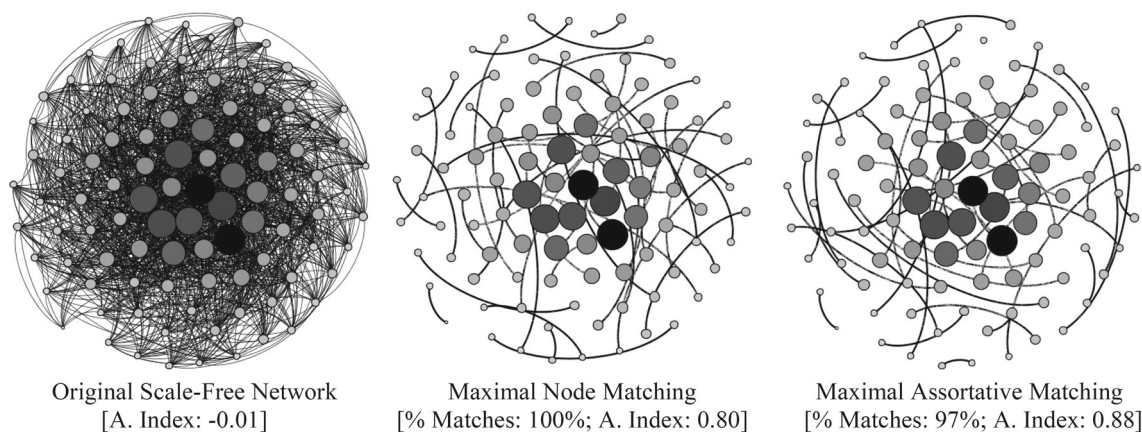


Fig. 40 Scale-free network (100 nodes)—BA model: initial—20 nodes; 20 links added per node introduction [spectral radius ratio for node degree: 1.13]

of the nodes decreases (vindicated by relatively lower values for the spectral radius ratio for node degree, as in Figs. 35, 36, 37, 38, 39, 40), facilitating the two algorithms (especially, the MAM algorithm) to pair similar nodes (with respect to

node degree). The A.Index of the MAM is significantly larger than that of the MNM for scale-free networks that have a lower number of new links added per node introduction (as large as by a difference of 0.4); as we increase the number

of links added per node introduction, the A.Index of MNM approaches to that of the MAM. Though there is a tradeoff expected between A.Index and the % of node matches, the % of node matches incurred with the MAM is only about 3–9 % low compared to the % of node matches incurred with the MNM (the larger differences are observed when the number of new links added per node introduction is low).

7 Maximal dissortative matching

Though the focus and objective of this paper is to develop an algorithm to find a maximal matching whose assortative index is maximum (close to 1), in this section, we want to illustrate that the proposed maximal assortative matching (MAM) algorithm (of Sect. 2) can also be used to determine a maximal matching whose assortative index is minimum (close to -1). We refer to the problem of finding a maximal matching with minimum assortative index as the maximal dissortative matching (MDM) problem. The MAM algorithm has to be only slightly modified to determine an MDM: instead of preferring to include edges with a lower assortative weight (to maximize the assortative index of the maximal matching), we need to include the uncovered edge with the largest assortative weight (to minimize the assortative index of the maximal matching) in each iteration. We refer to the MAM algorithm with the above modification as the MDM algorithm. The definition of the assortative weight remains the same as before, that is, the assortative weight of an uncovered edge (u, v) is the product of the number of uncovered edges adjacent to (u, v) and the absolute value of the difference in the node weights for the end vertices u and v .

The pseudo-code for the MDM algorithm to minimize the assortative index is shown in Fig. 41. The sub routines *FindAssortativeWeights* and *RemoveEdges* remain the same as before (see Sect. 2). We repeated the simulations of Sects. 3–6 for the MDM algorithm and averaged the results as we did

before in these sections. The results are presented in Table 3 (for real-world network graphs) and in Figs. 42, 43, 44, 45 and 46 for the theoretical models-based complex networks. Since the maximal node matching (MNM) algorithm works independent of the node weights, we do not show any comparison of the MDM algorithm with the MNM algorithm. The results presented in the earlier sections comparing the MAM (for maximizing the assortative index) with that of the MNM algorithm and the results presented in this section (comparing the MDM and MAM) would be sufficient to draw conclusions about the relative performance of the MDM vis-a-vis the MNM.

7.1 Analysis for real-world network graphs

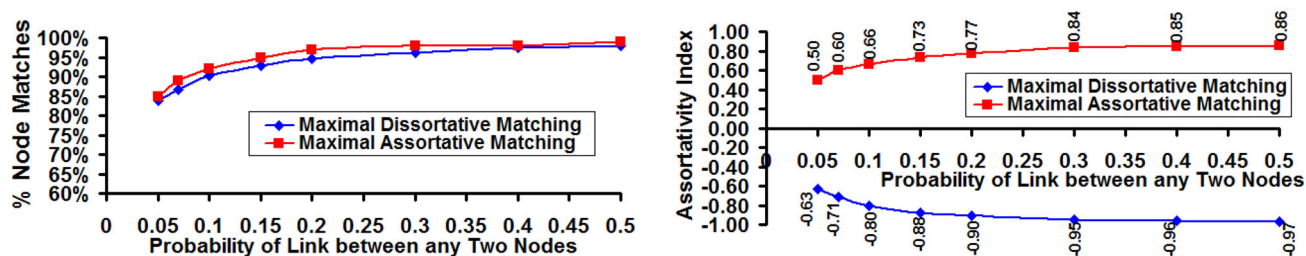
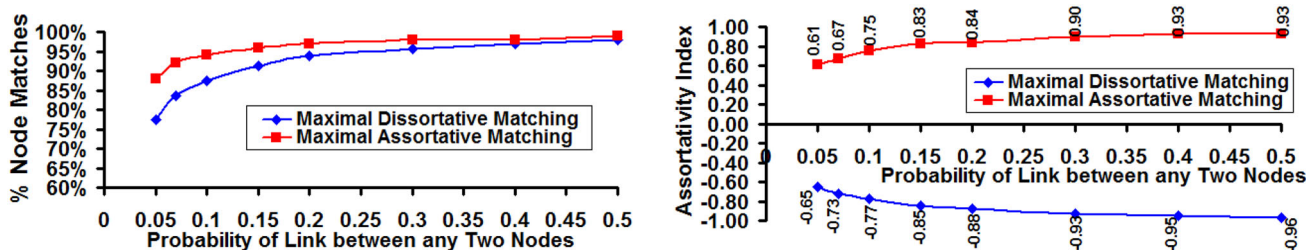
The results presented in Table 3 illustrate that for four of the six real-world network graphs, the maximal dissortative matching algorithm is not effective as the maximal assortative matching algorithm in optimizing the assortative index (A.Index). Though the assortative index values for the MDM for each of the six real-world network graphs are negative, the A.Index values for four of the six network graphs (US College Football Network, Dolphins' Social Network, US Politics Books Network, US Airports Network) are not that close to the optimal value of -1 compared to the proximity of the A.Index values observed for the MAM to the optimal value of 1. For the Karate Club Network and the Word Adjacencies Network, the assortativity index values observed for the MDM are relatively more closer or at the same distance to the targeted optimal value (-1) vis-a-vis the MAM to the targeted optimal value of 1. Except the Dolphins' Social Network for which the Maximal Dissortative Matching sustained a % of node matches that is 10 % larger than that incurred with maximal assortative matching, for all the other five real-world network graphs, the difference in the % of node matches between the two maximal matching strate-

Fig. 41 Pseudo-code for the maximal dissortative matching (MDM) algorithm

Input: Graph $G = (V, E)$, where weight $w(v) \in \mathbb{R}$ for every $v \in V$
Output: Maximal Dissortativity Matching, MDM
Auxiliary Variables: *UncoveredEdges*
Initialization: *UncoveredEdges* = E ; $MDM = \phi$
Begin MDM Algorithm
1 *FindAssortativityWeights*(*UncoveredEdges*)
2 **while** (*UncoveredEdges* $\neq \phi$) **do**
3 Edge $(u, v) = \left\{ (p, q) \mid \underset{(p, q) \in \text{UncoveredEdges}}{\text{Max}} [\text{AssortativityWeight}(p, q)] \right\}$
4 $MDM = MDM \cup \{(u, v)\}$
5 *UncoveredEdges* = *UncoveredEdges* - $\{(u, v)\}$
6 *RemoveEdges*((u, v) , *UncoveredEdges*)
7 *FindAssortativityWeights*(*UncoveredEdges*)
8 **end while**
9 **return** MDM
End MDM Algorithm

Table 3 Analysis of real-world networks for maximal assortative and maximal assortative matching

#	Real-world network	Network A.Index	Maximal assortative matching (MAM)		Maximal assortative matching (MDM)	
			% Node matches	A.Index	% Node matches	A.Index
(i)	US College Football Net.	−0.04	95	0.81	93	−0.48
(ii)	Dolphins' Social Net.	−0.04	73	0.82	83	−0.78
(iii)	US Politics Books Net.	−0.02	86	0.71	84	−0.51
(iv)	Karate Club Net.	−0.48	71	−0.13	70	−0.56
(v)	Word Adjacencies Net.	−0.10	78	0.50	79	−0.50
(vi)	US Airports 1997 Net.	−0.21	68	0.87	66	−0.24

**Fig. 42** Random networks: comparison of maximal dissortative matching and maximal assortative matching [node degree as node weights]**Fig. 43** Random networks: comparison of maximal dissortative matching and maximal assortative matching [random node weights]

gies (MAM and MDM) is only within a $\pm 2\%$ difference. The study conducted here could be used as a framework to decide which of the two maximal matching strategies (maximal assortative matching or maximal dissortative matching) would be relatively more effective/optimal (with respect to the proximity of the assortative index to the targeted optimal value) for a real-world network graph and accordingly the particular matching strategy could be applied.

7.2 Analysis for random network graphs

The results presented for random network graphs with node degree as node weights (Fig. 42) illustrate that the assortative index values obtained with maximal dissortative matching (MDM) is more close to the targeted optimal value (-1) compared to the closeness of the assortative index values obtained with the maximal assortative matching (MAM) to the targeted optimal value (1). On the other hand, though the % of node matches obtained with maximal dissortative

matching appears to be less than that obtained with the maximal assortative matching, the difference in the % of node matches is within 2–3 % for all values of p_{link} and by observing the nature of the increase in the % of node matches with the two maximal matching strategies, we could say that the difference in the % of node matches would only further narrow down with increase in the p_{link} value. The results of Fig. 42 thus illustrate that for random network graphs with node degree as node weights, it would be more apt to target a maximal dissortative matching compared to a maximal assortative matching on the basis of the proximity of the assortative index to the targeted optimal value (-1 for MDM and $+1$ for MAM).

When we run the MDM algorithm on random network graphs (that evolved using the Erdos–Renyi model) with randomly generated node weights in the range $(0, \dots, 1]$, we observe (from Fig. 43) the assortativity index values for the maximal dissortative matching to be very close to that of the assortativity index values illustrated in Fig. 42 for the

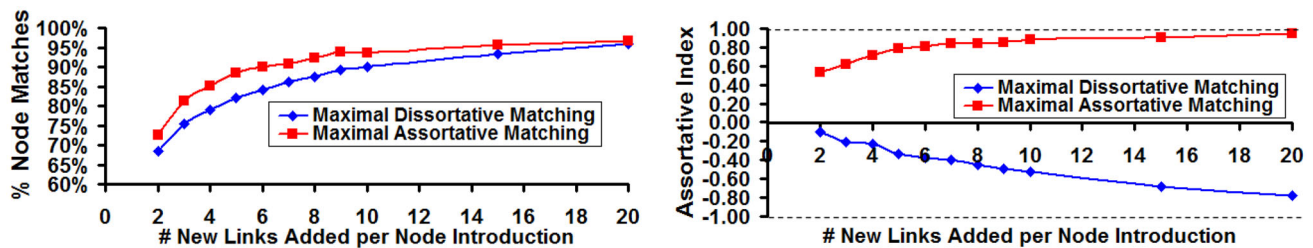


Fig. 44 Scale-free networks: comparison of maximal dissortative matching and maximal assortative matching (initial # nodes: 3; total nodes: 100)

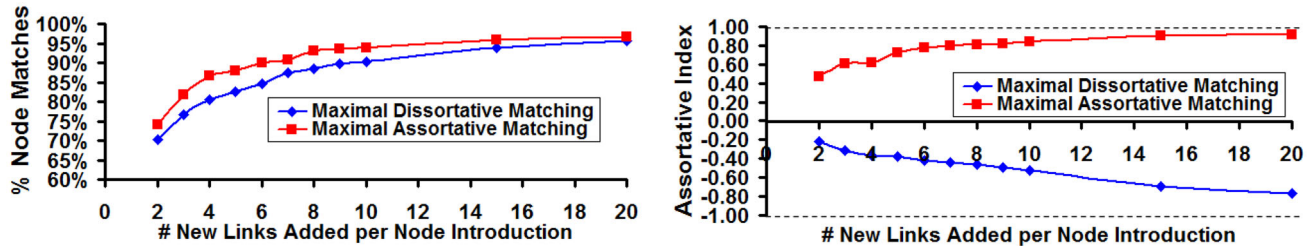


Fig. 45 Scale-free networks: comparison of maximal dissortative matching and maximal assortative matching (initial # nodes: 10; total nodes: 100)

maximal dissortative matching obtained on random network graphs with node degree as node weights (the difference in A.Index is within ± 0.03); the % of node matches obtained for the maximal dissortative matching with random node weights is at most 7 % lower than that obtained for the maximal dissortative matching with node degree as node weights.

While comparing the results obtained for the maximal assortative matching and maximal dissortative matching obtained on random network graphs with random node weights, we observe the assortativity index values of the maximal dissortative matching to be relatively more closer to the targeted optimal value of -1 compared to that of the closeness of the assortativity index values of the maximal assortative matching to its targeted optimal value of 1 . Thus (like in the case of random network graphs with node degree as node weights), we could still say that for random network graphs with random node weights, it would be more apt to aim for a maximal dissortative matching compared to a maximal assortative matching on the basis of the proximity of the assortativity index to the targeted optimal value.

7.3 Analysis for scale-free network graphs

Figures 44, 45 and 46 present the results of the execution of the MDM and MAM algorithms on scale-free network graphs that evolve from the BA model (described in Sect. 6); the node degree is used as node weights for the assortativity calculations. The simulation conditions are the same as those used in Sect. 6: the values for the initial number of nodes (n_{init}) are 3, 10 and 20; the values for the number of new links added per node introduction ($links_{\text{new}}$) are: 2, 3, 4, 5, 6, 7, 8, 9, 10, 15 and 20. The results presented in

Figs. 44, 45 and 46 are the average of 100 trial runs of the simulations for each of the above combinations of the n_{init} and $links_{\text{new}}$ values. Overall, for a given operating condition ($n_{\text{init}}, links_{\text{new}}$), we observe the assortativity index values for a maximal assortative matching to be relatively more closer to the targeted optimal value of 1 vis-a-vis the closeness of the assortativity index values for a maximal dissortative matching to the targeted optimal value of -1 . Likewise, the % of node matches observed with the maximal assortative matching is slightly larger than that obtained with the maximal dissortative matching for all operating conditions (the difference could be at most 7 %). For a given n_{init} , the % of node matches between the two maximal matching strategies is larger at lower values of $links_{\text{new}}$ and the difference narrows down as we increase the value of $links_{\text{new}}$. Thus, overall, maximal assortative matching would be relatively more apt for scale-free networks with respect to the proximity towards the targeted optimal value for the assortativity index vis-a-vis the maximal dissortative matching (1 for MAM; -1 for MDM).

With regard to the nature of increase or decrease with respect to the each of the two operating parameters, we observe the following: For a given value of $links_{\text{new}}$, as we increase the n_{init} value, the values for the assortativity index for a maximal dissortative matching get more closer to the targeted optimal value of -1 . This is contrary to what has been observed for maximal assortative matching: for a given value of $links_{\text{new}}$, as we increase the n_{init} value, the values for the assortativity index for a maximal assortative matching move farther away from the targeted optimal value of 1 . On the contrary, for a given n_{init} value, as we increase $links_{\text{new}}$, we observe the assortativity index values for the maximal assortative matching to get relatively more

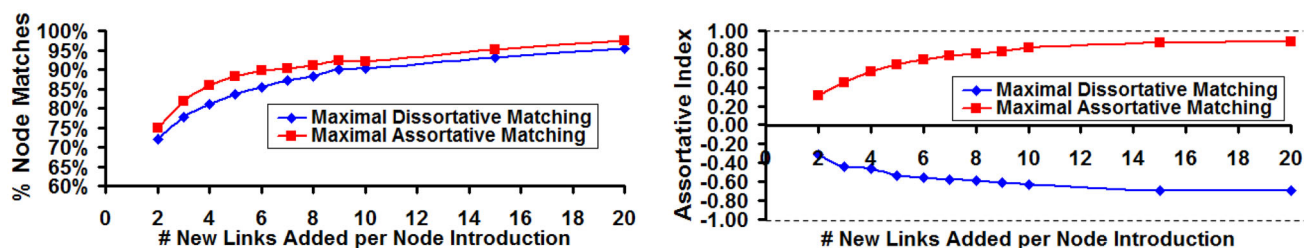


Fig. 46 Scale-free networks: comparison of maximal dissortative matching and maximal assortative matching (initial # nodes: 20; total nodes: 100)

closer to the targeted optimal value (1) compared to what is observed for the assortativity index values for the maximal dissortative matching with respect to the target optimal value (−1).

7.4 Correlation with the degree distribution for real-world networks and scale-free networks

An interesting observation from the results presented in Figs. 44, 45, 46 and Table 3 is that the larger the magnitude of the difference in the assortativity index (A.Index) values for the MAM and MDM (i.e., $A.Index_{MAM} - A.Index_{MDM}$) with respect to node degree, the smaller the spectral radius ratio for node degree (i.e., the smaller the variation in the node degree) for the corresponding network and vice-versa. For example, in Table 3, the magnitude of difference in the assortativity index values between the MAM and MDM of the US College Football network (with a spectral radius ratio for node degree 1.01) is $0.81 - (-0.48) = 1.29$; on the other hand, for the Karate Club network (with a spectral radius ratio for node degree 1.46), the magnitude of the difference between the assortativity index values for the MAM and MDM is $-0.13 - (-0.56) = 0.43$.

For a given initial number of nodes (n_{init}) during the evolution of a scale-free network, we could observe that the difference in the assortative index values for the MAM and MDM gets larger with increase in the values for the number of new links ($links_{new}$) added per node introduction (which leads to a decrease in the spectral radius ratio for node degree). To get a better understanding of this relationship, we compiled the results observed for the assortative index (A.Index) of MAM and MDM obtained for scale-free networks with different n_{init} and $links_{new}$ values used in the simulations, and plotted the difference ($A.Index_{MAM} - A.Index_{MDM}$) vs. the Spectral radius ratio for node degree in a single plot: we observe an inverse relationship (see Fig. 47) between the spectral radius ratio for node degree and the difference between the assortativity index values for MAM and MDM; the correlation coefficient is -0.90 . Note that the larger the difference between the A.Index values for MAM and MDM, the more closer the A.Index values of the respective maximal matching to their targeted optimal values (−1

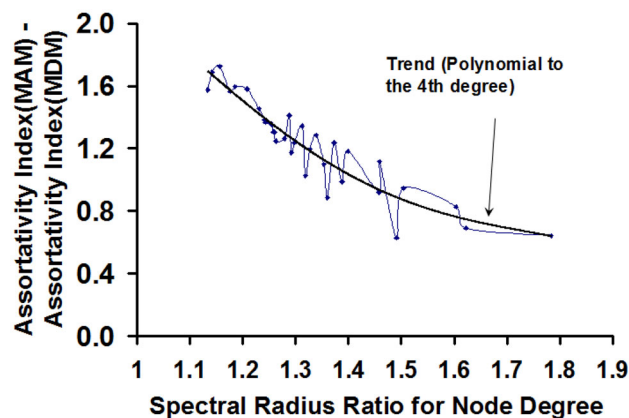


Fig. 47 Scale-free networks: correlation between spectral radius ratio for node degree and the difference between the assortativity index values for maximal assortative matching (MAM) and maximal dissortative matching (MDM)

for MDM and 1 for MAM) and vice-versa. Hence, the correlation discussed here between the spectral radius ratio for node degree and the difference between the A.Index values for the maximal a(di)ssortative matching indicate that for scale-free networks with a larger variation in node degree, it is less likely for the assortative index of a maximal assortative or maximal dissortative matching to be closer to the targeted optimal value, and vice-versa.

In the case of theoretically generated random networks (from the Erdos–Renyi model), since the spectral radius ratio for node degree is more likely to be close to its minimum value of 1.0 and less variation is expected among the nodes with respect to degree, we do not attempt to correlate the spectral radius ratio for node degree and the difference in the assortativity index values between the MAM and MDM for random networks.

8 Related work

Before the surge in interest for social network analysis, the graphs considered for maximum matching are typically bipartite graphs wherein there exists two sets of vertices (with no edges between vertices in the same set) and the edges con-

nect the vertices from one set to the other set. Given a bipartite graph with no edge weights, the maximum matching problem would be about determining the maximum number of matches between the vertices across the two sets of the graph. If the edges of a bipartite graph have weights, the maximum matching problem would be about determining the set of matching edges (no two edges in the set have overlapping vertices) such that the sum of the edge weights is the maximum. The maximum matching problem for bipartite graphs could be optimally solved using well-known polynomial-time algorithms such as the Edmonds–Karp algorithm [22].

The maximal assortative matching problem has been so far not considered in the literature for bipartite graphs. Instead, a related problem, called the stable matching problem, was considered for bipartite graphs and is defined as follows: given a set of preferences for each vertex of the two partitions of a bipartite graph, a matching of the vertices from one partition to another partition is considered to be stable if there does not exist any pair of vertices (A, B) such that A is matched to some other vertex that is less preferred than B and likewise, B is matched to some other vertex that is less preferred than A . The Gale–Shapley algorithm [23] is a well-known algorithm for stable matching in bipartite graphs with an equal number of vertices in both the partitions. We do not see any possible extension of this algorithm or any other stable matching algorithm proposed for bipartite graphs to determine maximal assortative or maximal disassortative matching for arbitrary network graphs.

If a maximum matching is needed for directed network graphs, the common strategy in the literature is to get the bipartite equivalent of the network graph and apply the Edmonds–Karp or any other algorithm for determining maximum matching in bipartite graphs. The problem of determining the bipartite equivalent for a directed graph is an NP-hard problem [24]. A well-known heuristic using clique covering has been proposed in [25] for transforming a directed graph to a bipartite graph. In [26], an alternate strategy was proposed using the concept of structural controllability [27] to determine maximum matching in directed complex network graphs, bypassing the need to first transform to a bipartite graph. This algorithm is targeted at maximizing the number of nodes that are part of a matching and is not designed to maximize or minimize the assortativity index.

In [28], the authors showed that for networks with binomial degree distribution, the maximum and minimum assortativity vary with the density of the networks. Motivated by this observation, the authors in [29] introduced an algorithm to compute a network with maximal or minimal assortativity given a vector of valid node degrees using degree-preserving rewiring [30] and weighted b-matching [31]. Degree-preserving link rewiring is effective in decreasing or increasing the assortativity of a network graph without

affecting the degree distribution of the vertices. However, neither the work in [28] nor in [29] could be extended to determine a maximal a(di)ssortative matching of the edges of the graph. It was also shown in [28] that for networks whose degree distribution is binomial (like the Erdos–Renyi model-based random network graphs), the maximum assortativity and minimum assortativity are asymptotically anti-symmetric. This observation correlates well with our observation in Sect. 7 that the values for the assortative index for maximum assortative matching are comparable enough to the absolute values of the assortative index for maximum disassortative matching, especially in the case of the random network graphs with random node weights as well as with node degree as node weights.

In [5, 32], Piraveenan et al. explore degree assortativity in complex networks and propose that a perfect degree assortativity is possible if the network could be fragmented into sub networks, whereby each sub network is a complete network; on the other hand, perfect degree assortativity has been considered to be relatively more difficult to achieve in complex networks, except the case of complete bipartite graphs [33] (like a star graph). Even though perfect degree assortativity and perfect degree disassortativity are difficult to be observed in all kinds of complex networks, in this paper, we show that it is possible to find a matching of the vertices such that the assortative index is significantly close to the optimal value (especially in the case of maximal assortative matching).

In [34], the authors repeatedly employed degree-preserving link rewiring on a given complex network graph (generated from a theoretical model) to obtain an ensemble of graphs and measured the range of values for the assortativity and clustering coefficient for the ensemble of graphs; the broader the range of the values for the assortativity and clustering coefficient, the narrower the degree distribution of the original graph and vice-versa. In Sect. 7.4, we discuss the correlation between the difference in the assortativity index values (calculated based on node degree) for the MAM and MDM vis-a-vis the spectral radius ratio for node degree in a scale-free network. We show that instead of additionally considering clustering coefficient (as in [34]), the assortativity index values of the maximal assortative matching and maximal disassortative matching alone could be used to characterize the variation in the degree distribution for scale-free networks.

The problem of determining a maximal matching with minimum cardinality for the set of edges constituting the matching is an NP-hard problem [35]. It is equivalent to the problem of finding a minimum edge dominating set [36]—to find the smallest set of edges of the graph such that each edge in the set covers itself and covers one or more adjacent edges as well as satisfies the matching constraint (no two edges in the set have a common end vertex). The

computational time-complexity of the heuristics [35–37] to determine approximations to the minimum edge dominating set is $O((E + V)\log E)$ for a graph of V vertices and E edges. Similarly, another related problem: the problem of determining a connected dominating set of minimum size (to find the smallest set of connected vertices such that each vertex in the set covers itself and one or more of its adjacent vertices) is also a NP-hard problem. Efficient heuristics [38,39] of computational-time complexity $O((E + V)\log V)$ have been proposed in the literature to determine approximations to minimum connected dominating sets for complex network graphs. Though the heuristics for both the minimum edge dominating set and minimum node connected dominating set problems have a computational time-complexity that is smaller than our proposed $O(EV\log V)$ heuristic for maximal assortative matching, these heuristics cannot be applied for the problem of focus in this paper.

The problem of focus in this paper is the maximal independent edge set problem [7] wherein we want to find the largest set of independent edges such that no two edges have a common end vertex. Note that heuristics (e.g., [37]) for the minimum edge dominating set problem cannot be applied to determine the maximal node matching and the maximal a(di)ssortative matching because heuristics for the minimum edge set problem are more likely to determine the set of edges such that each edge in the set covers a larger number of adjacent edges. Similarly, heuristics for the minimum connected dominating set problem prefer to include nodes that could cover several adjacent nodes (and the associated edges). Hence, using the heuristics for minimum edge dominating set or minimum connected dominating set for determining a maximal node matching or maximal assortative matching of the edges would only reduce the number of independent edges that become part of the matching. The maximal matching algorithms developed in this paper take the approach of preferring to include edges that cover a smaller number of adjacent edges so that the number of independent edges determined could be as large as possible. To the best of our knowledge, we have not come across a maximal matching algorithm that is aimed at simultaneously maximizing the a(di)ssortativity of the matching as well as maximizing the cardinality of the matching for complex network graphs. In this perspective, the maximal assortative matching and maximal dissortative matching algorithms proposed in this paper are significant contributions to the literature for complex network graphs and analysis.

9 Conclusions

The results of the execution of the maximal assortative matching (MAM), maximal dissortative matching (MDM)

and maximal node matching (MNM) algorithms on the complex network graphs generated from theoretical models as well as on the real-world network graphs convey useful insights. We observe that the MAM and MDM algorithms could be, respectively, used to determine maximal assortative matching and maximal dissortative matching (matching nodes of similar weights or dissimilar weights, depending on the application) for various complex network graphs (including social networks) without any significant loss in the % of node matches vis-a-vis the maximal node matching. On the other hand, we observe the assortative index of a maximal node matching to be far away from the targeted optimal values of -1 and 1 (indicating that maximal node matching is more arbitrary with respect to the pairing of the vertices); however, such an arbitrary matching is of no use for networks that require the users (nodes) to be matched to other nodes of similar or dissimilar weights, as in the case of social networks. In the case of the complex network graphs generated from theoretical models, we have also identified which of the two maximal matching strategies (MAM or MDM) are likely to incur an assortativity index that is closer to their targeted optimal values (1 for MAM and -1 for MDM). We observe the random network graphs (generated from the Erdos–Renyi model) to be more apt for a maximal dissortative matching (MDM) and the scale-free network graphs (generated from the Barabasi–Albert model) to be more apt for a maximal assortative matching (MAM). We also observe that the difference between the assortative index values for an MAM and MDM is negatively correlated to the variation in the node degree for scale-free networks generated according to the BA model as well as for the real-world network graphs studied in this paper.

Open Access This article is distributed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits unrestricted use, distribution, and reproduction in any medium, provided you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license, and indicate if changes were made.

References

1. Edmonds, J.: Paths, trees, and flowers. *Can. J. Math.* **17**, 449–467 (1965)
2. Micali, S., Vazirani, V.: An $O(V^{1/2}E)$ Algorithm for Finding Maximum Matching in General Graphs. In: *Proceedings of the 21st Annual Symposium on Foundations of Computer Science*, pp. 17–27, Syracuse, NY, USA (1980)
3. Newman, M.E.J.: Assortative mixing in networks. *Phys. Rev. Lett.* **89**, 208701 (2002)
4. Piraveenan, M., Prokopenko, M., Zomaya, A.: Local assortativeness in scale-free networks. *Europhys. Lett.* **84**(2), 28002 (2008)
5. Piraveenan, M., Prokopenko, M., Zomaya, A.: Assortative mixing in directed biological networks. *IEEE/ACM Trans. Comput. Biol. Bioinform.* **9**(1), 66–78 (2012)

6. Newman, M.E.J.: *Networks: An Introduction*, 1st edn. Oxford University Press, Oxford (2010)
7. Cormen, T.H., Leiserson, C.E., Rivest, R.L., Stein, C.: *Introduction to Algorithms*, 3rd edn. MIT Press, Cambridge (2009)
8. Strang, G.: *Linear Algebra and Its Applications*, 4th edn. Brooks Cole, Pacific Grove (2006)
9. Caldarelli, G.: *Scale-Free Networks: Complex Webs in Nature and Technology*, 1st edn. Oxford University Press, Oxford (2007)
10. Erdos, P., Renyi, A.: On random graphs. *Publ. Math.* **6**, 290–297 (1959)
11. Barabasi, A.-L., Albert, R.: Emergence of scaling in random networks. *Science* **286**(5439), 509–512 (1999)
12. Demange, M., Ekim, T.: Minimum maximal matching is NP-hard in regular bipartite graphs. *Theory and Applications of Models of Computation. Lecture Notes in Computer Science*, vol. 4978, pp. 364–374 (2008)
13. Girvan, M., Newman, M.: Community structure in social and biological networks. *Proc. Natl. Acad. Sci. U. S. A.* **19**(12), 7821–7826 (2002)
14. Lusseau, D., Schneider, K., Boisseau, O.J., Haase, P., Slooten, E., Dawson, S.M.: The bottlenose dolphin community of Doubtful Sound features a large proportion of long-lasting associations. *Behav. Ecol. Sociobiol.* **54**, 396–405 (2003)
15. Krebs, V.: Working in the connected world: book network. *J. Inst. Health Rec. Inf. Manag.* **4**(1), 87–90 (2000)
16. Zachary, W.W.: An information flow model for conflict and fission in small groups. *J. Anthropol. Res.* **33**(4), 452–473 (1977)
17. Newman, M.: Finding community structure in networks using the eigenvectors of matrices. *Phys. Rev. E* **74**, 036104 (2006)
18. Pajek Datasets: <http://vlado.fmf.uni-lj.si/pub/networks/data/map/USAir97.net>. Last accessed 25 May 2015
19. Meghanathan, N.: Spectral Radius as a Measure of Variation in Node Degree for Complex Network Graphs. In: *Proceedings of the 3rd International Conference on Digital Contents and Applications*, pp. 30–33, Hainan, China (2014)
20. Cherven, K.: *Network Graph Analysis and Visualization with Gephi*, 1st edn. Packt Publishing, Birmingham (2013)
21. Fruchterman, T.M.J., Reingold, E.M.: Graph drawing by force-directed placement. *Softw. Pract. Exp.* **21**(11), 1129–1164 (1991)
22. Edmonds, J., Karp, R.M.: Theoretical improvements in algorithmic efficiency for network flow problems. *J. ACM* **19**(2), 248–264 (1972)
23. Shoham, Y., Leyton-Brown, K.: *Multiagent Systems: Algorithmic, Game-Theoretic, and Logical Foundations*, 1st edn. Cambridge University Press, Cambridge (2009)
24. Kalman, R.E.: Mathematical description of linear dynamical systems. *J. Soc. Ind. Appl. Math. Ser. A Control* **1**(2), 152–192 (1963)
25. Guillaume, J.-L., Latapy, M.: Bipartite graphs as models of complex networks. *Physica A Stat. Mech. Appl.* **371**(2), 795–813 (2006)
26. Chatterjee, A., Das, D., Naskar, M.K., Pal, N., Mukherjee, A.: Heuristic for Maximum Matching in Directed Complex Networks. In: *Proceedings of the International Conference on Advances in Computing, Communications and Informatics*, pp. 1146–1151 (2013)
27. Liu, Y.-Y., Slotine, J.-J., Barabasi, A.-L.: Controllability of complex networks. *Nature* **473**, 167–173 (2011)
28. Wang, H., Winterbach, W., Van Mieghem, P.: Assortativity of complementary graphs. *Eur. Phys. J. B* **83**(2), 203–214 (2011)
29. Winterbach, W., de Ridder, D., Wang, H.J., Reinders, M., Van Mieghem, P.: Do greedy assortativity optimization algorithms produce good results? *Eur. Phys. J. B* **5**, 151–160 (2012)
30. Maslov, S., Sneppen, K.: Specificity and stability in topology of protein networks. *Science* **296**(5569), 910–913 (2002)
31. Muller-Hannemann, M., Schwartz, A.: Implementing weighted b-matching algorithms: insights from a computational study. *Algorithm Engineering and Computation. Lecture Notes in Computer Science*, vol. 1619, pp. 18–36 (1999)
32. Piraveenan, M., Prokopenko, M., Zomaya, A.: Assortativeness and information in scale-free networks. *Eur. Phys. J. B* **67**(3), 291–300 (2009)
33. Van Mieghem, P., Wang, H., Ge, X., Tang, S., Kuipers, F.: Influence of assortativity and degree-preserving rewiring on the spectra of networks. *Eur. Phys. J. B* **76**(4), 643–652 (2010)
34. Holme, P., Zhao, J.: Exploring the assortativity-clustering space of a network's degree sequence. *Phys. Rev. E* **75**, 046111 (2007)
35. Yannakakis, M., Gavril, F.: Edge dominating sets in graphs. *SIAM J. Appl. Math.* **38**(3), 364–372 (1980)
36. Horton, J.D., Kilakos, K.: Minimum edge dominating sets. *SIAM J. Discrete Math.* **6**(3), 375–387 (1993)
37. Cardinal, J., Labbe, M., Langerman, S., Levy, E., Melot, H.: A Tight Analysis of the Maximal Matching Heuristic. *Lecture Notes in Computer Science*, vol. 3595, pp. 701–709 (2005)
38. Meghanathan, N.: Centrality-based connected dominating sets for complex network graphs. *Int. J. Interdiscip. Telecommun. Netw.* **6**(2), 1–19 (2014)
39. Erciyes, K.: *Complex Networks: An Algorithmic Perspective*, 1st edn. CRC Press, Boca Raton (2014)